SiM3L1xx Reference Manual

This reference manual accompanies several documents to provide the complete description of SiM3L1xx devices, part of the Silicon Laboratories 32-bit ARM Cortex-M3 family of microcontrollers.

This document provides the detailed description for all peripherals available on all SiM3L1xx devices. The peripheral mix varies across different members of the device family. Refer to the device data sheet for details on the specific peripherals available for each member of the device family. In the event that the device data sheet and this document contain conflicting information, the device data sheet should be considered the authoritative source.

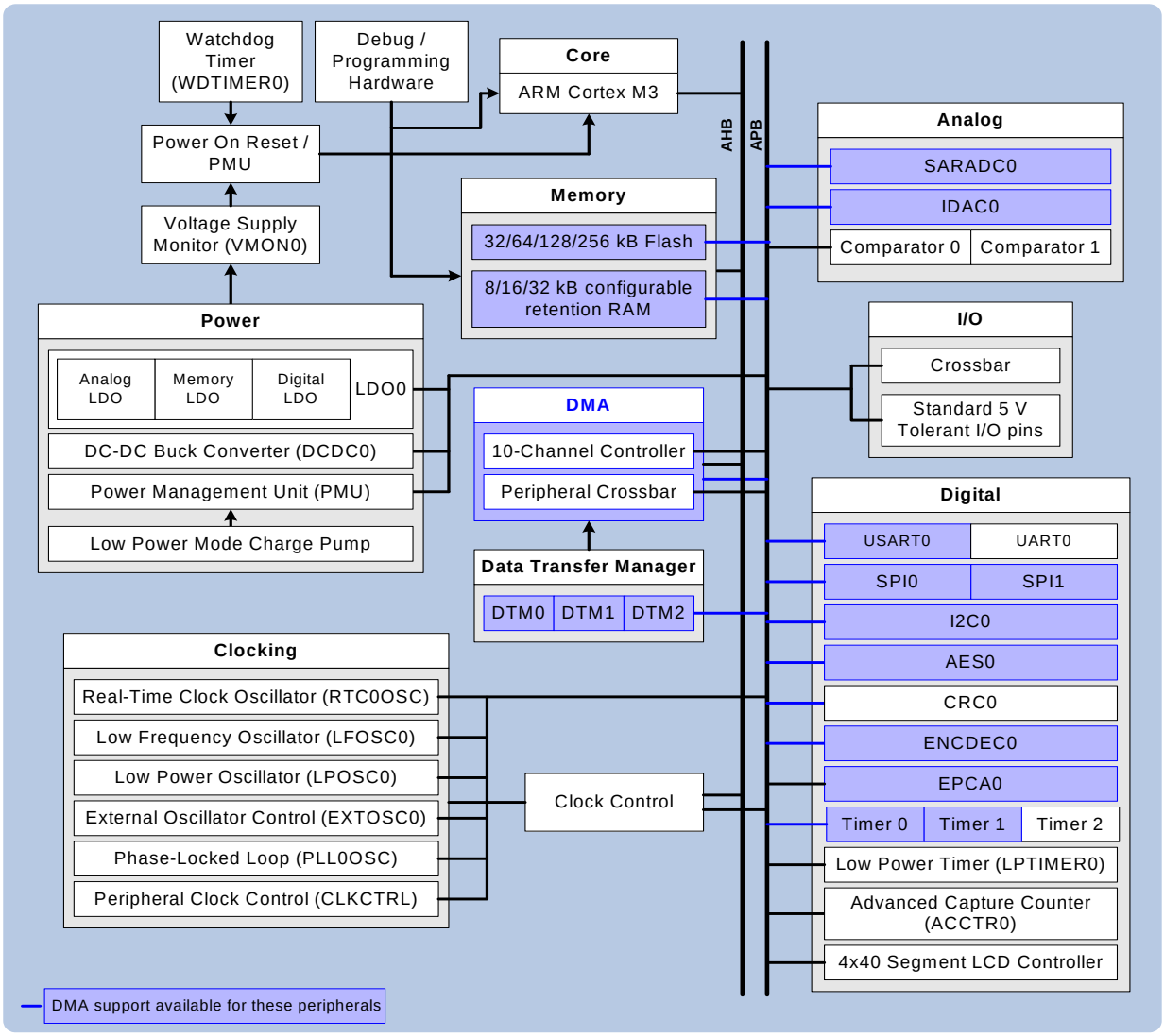


Table of Contents

1. Related Documents and Conventions

1.1. Related Documents

1.1.1. SiM3L1xx Data Sheet

The Silicon Laboratories SiM3L1xx Data Sheet provides specific information for this device family, including electrical characteristics, mechanical characteristics, and ordering information.

1.1.2. Hardware Access Layer (HAL) API Description

The Silicon Laboratories Hardware Access Layer (HAL) API provides functions to modify and read each bit in the SiM3L1xx devices. This description can be found in the SiM3xxxx HAL API Reference Manual.

1.1.3. ARM Cortex-M3 Reference Manual

The ARM-specific features like the Nested Vector Interrupt Controller are described in the ARM Cortex-M3 reference documentation. The online reference manual can be found online at the following link: http://infocenter.arm.com/help/topic/com.arm.doc.subset.cortexm3/index.html#cortexm3.

1.2. Conventions

The block diagrams in this document use the following formatting conventions:

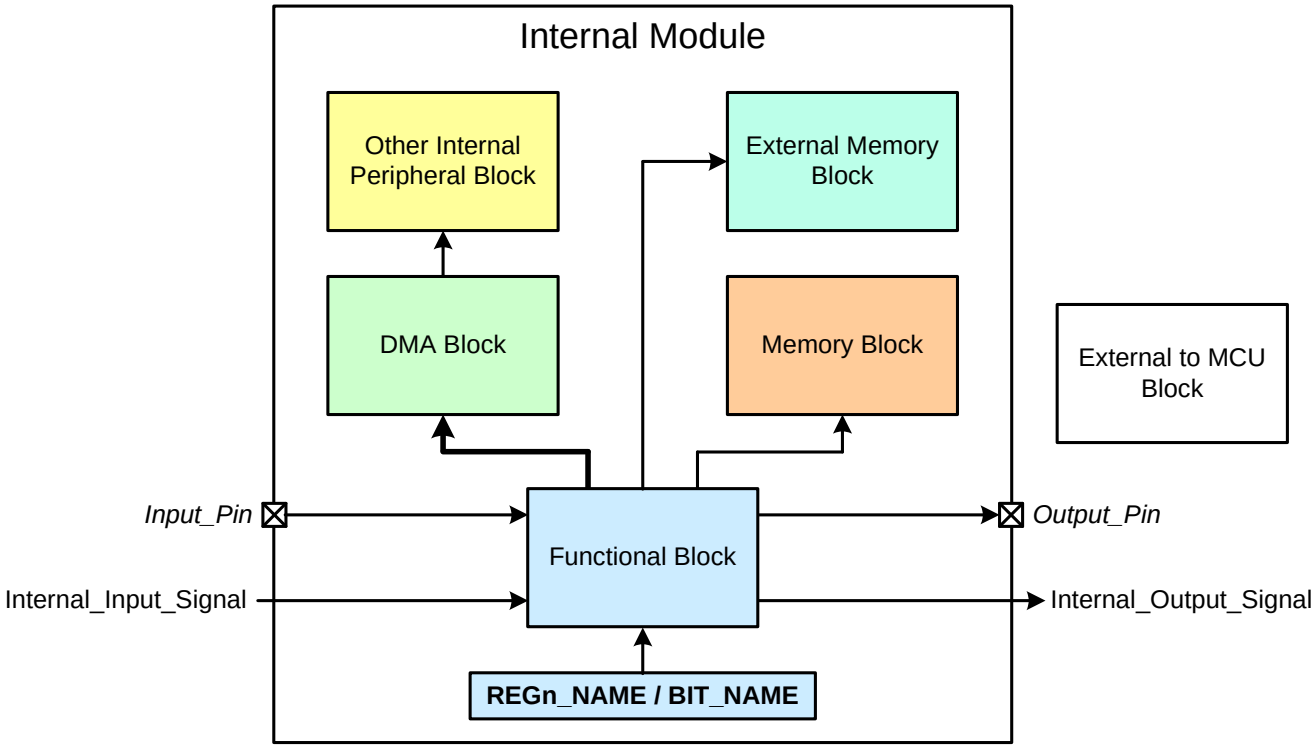


Figure 1.1. Block Diagram Conventions

2. Memory Organization

The memory organization of the SiM3L1xx devices follows the standard ARM Cortex-M3 structure, shown in Figure 2.1. There is one 32-bit memory space shared amongst the Flash, RAM, SiM3L1xx Peripherals, External Memory, and M3 Peripherals. The unused memory addresses are reserved and should not be accessed.

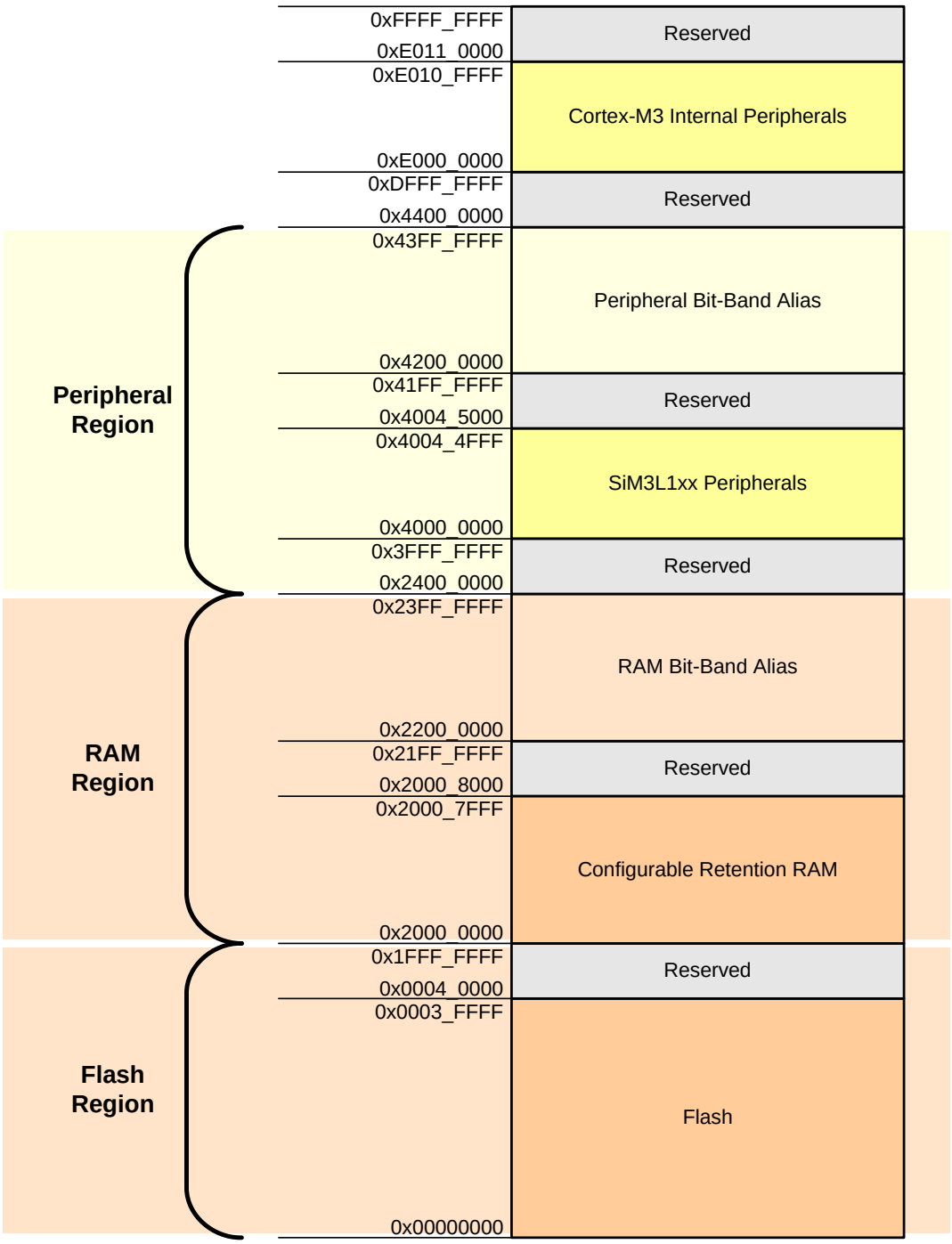


Figure 2.1. SiM3L1xx Memory Map

2.1. Flash Region

The SiM3L1xx devices implement 256, 128, 64, or 32 kB of Flash which is accessible starting at 0x00000000. The Flash can be read using standard ARM instructions. The FLASHCTRL0 module should be used to write and erase Flash from firmware.

The Flash block can be locked from external access by writing to the lock word located at 0x0003FFFC. A value of 0xFFFFFFFF or 0x00000000 at this location will unlock the Flash. Any other value written to this location will lock the entire Flash from external (debugger) writes or reads until:

- An erase operation is initiated from firmware.
- An erase operation is initiated through the debug port (SWD/JTAG).
- Firmware writes 0x00000000 to the lock word.

The DMA can access all of Flash.

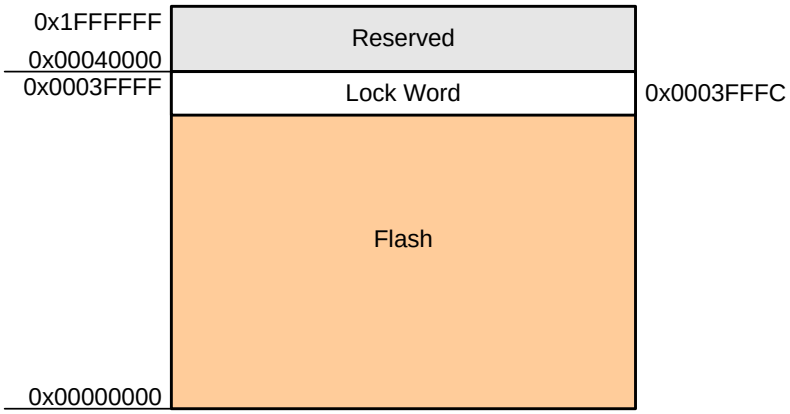


Figure 2.2. SiM3L16x Flash Memory Map (256 kB)

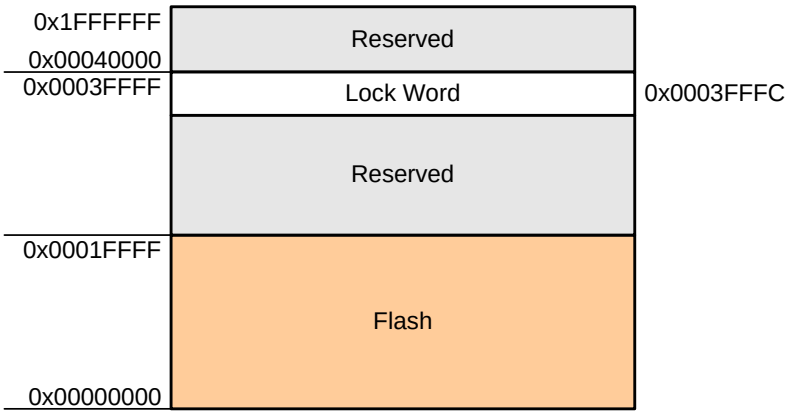


Figure 2.3. SiM3L15x Flash Memory Map (128 kB)

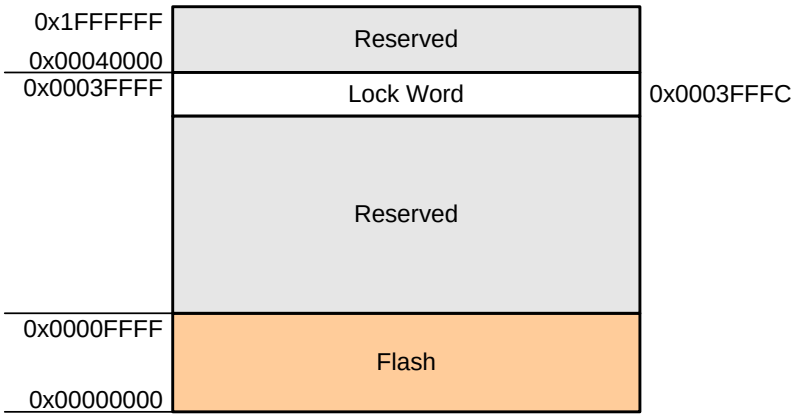


Figure 2.4. SiM3L14x Flash Memory Map (64 kB)

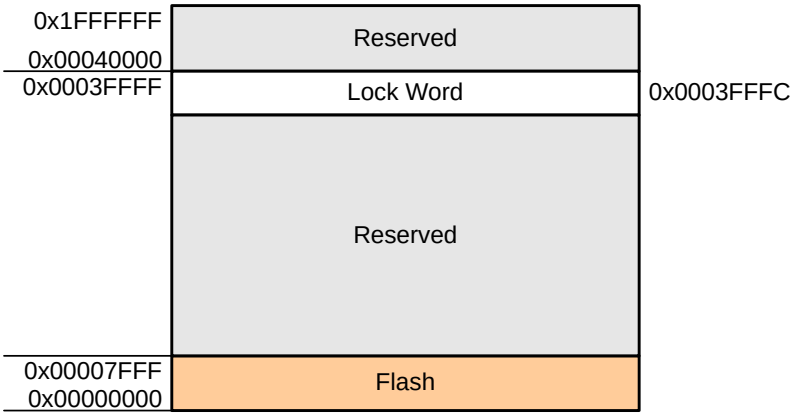


Figure 2.5. SiM3L13x Flash Memory Map (32 kB)

2.2. RAM Region

The RAM region consists of 32 kB (SiM3L16x and SiM3L15x), 16 kB (SiM3L14x), or 8 kB (SiM3L13x) and starts at location 0x20000000. This RAM is configurable via firmware in 4 kB segments to act as standard RAM or retention RAM in the PMU0 module. When a 4 kB block is configured as retention RAM, memory contents will be retained during Power Mode 8 and 9 as long as the Supply Monitor has not caused a reset.

The RAM Bit-Band Alias region can be used to perform sets or clears of individual bits in the RAM. Each bit in the RAM region is represented by the least-significant bit at the word-aligned bit-band alias address.

2.3. Peripheral Region

The SiM3L1xx peripheral registers are located starting at address 0x4000_0000. Registers for a specific module are typically located together in the peripheral region of memory to facilitate structure access from firmware. Each register may have up to four access methods, implemented as four separate locations in memory. The four possible access methods are named ALL, SET, CLR, and MSK.

The register's ALL access address is the primary access point for any register. Individual bits may be Read/Write (RW), Read-Only (RO), or Write-Only (WO). The ALL access address is implemented for all registers, and where absolute memory addresses are given in the documentation, they refer to the ALL address. For registers with write access, the ALL address will directly write all bits of the register. A read of the ALL address will read the current value in the register.

The SET and CLR addresses provide bit-wise, atomic write access to set and clear bits in the register without colliding with hardware. Writing a 1 to a bit in the SET address will set the corresponding bit, and writing a 1 to a bit in the CLR address will clear the corresponding bit. A write of 0 to either SET or CLR will have no effect on the corresponding bit. For registers implementing SET and CLR access methods, the SET address is at offset 0x4, and the CLR address is at offset 0x8 from the register's ALL access address. SET and CLR access are not implemented on every register.

The MSK address allows a write to a specific range of bits in the register. The upper 16 bits act as a mask for writing a value in the lower 16 bits of the register. For example, a write of 0x0F000400 to the MASK address would write a value of 4 to bits [11:8] of the register, while none of the rest of the bits are modified. For registers implementing the MSK access method, the MSK address is at offset 0xC from the registers ALL access address. MSK access is implemented for only a small set of registers which may require atomic, simultaneous writes of both 1's and 0's (such as port output registers).

Many control and status registers support the SET and CLR access methods. The Peripheral Bit-Band Alias region can also be used to perform sets or clears of individual bits in the peripheral registers, which results in a read-modify-write operation on the bus. Each bit in the registers region is represented by the least-significant bit at the word-aligned bit-band alias address. When supported, it is recommended to use the SET and CLR registers instead of the Bit-Band Alias region to change individual bits in a register. Bit-band accesses are not protected against hardware conflicts.

Each peripheral is discussed in detail in the corresponding chapter. The register map for the SiM3L1xx devices can be found in "3. SiM3L1xx Register Memory Map" . Detailed descriptions of each register and the bit fields within can be found in the specific peripheral section for that register.

2.4. Cortex-M3 Internal Peripherals

The Cortex-M3 Internal Peripherals space includes standard M3 functions such as the NVIC and ETM. For more information on these functions of the ARM core, consult the ARM Cortex-M3 Reference Manual.

3. SiM3L1xx Register Memory Map

This section details the register memory map for the SiM3L1xx devices. Registers are listed in address order, beginning with 0x4000_0000.

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
USART0 Registers					
USART0_CONFIG	Module Configuration	0x4000_0000	Y	Y	
USART0_MODE	Module Mode Select	0x4000_0010	Y	Y	
USART0_FLOWCN	Flow Control	0x4000_0020	Y	Y	
USART0_CONTROL	Module Control	0x4000_0030	Y	Y	
USART0_IPDELAY	Inter-Packet Delay	0x4000_0040			
USART0_BAUDRATE	Transmit and Receive Baud Rate	0x4000_0050			
USART0_FIFOCN	FIFO Control	0x4000_0060	Y	Y	
USART0_DATA	FIFO Input/Output Data	0x4000_0070			
UART0 Registers					
UART0_CONFIG	Module Configuration	0x4000_1000	Y	Y	
UART0_MODE	Module Mode Select	0x4000_1010	Y	Y	
UART0_FLOWCN	Flow Control	0x4000_1020	Y	Y	
UART0_CONTROL	Module Control	0x4000_1030	Y	Y	
UART0_IPDELAY	Inter-Packet Delay	0x4000_1040			
UART0_BAUDRATE	Transmit and Receive Baud Rate	0x4000_1050			
UART0_FIFOCN	FIFO Control	0x4000_1060	Y	Y	
UART0_DATA	FIFO Input/Output Data	0x4000_1070			
UART0_CLKDIV	Clock Divider	0x4000_1080			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
SPI0 Registers					
SPI0_DATA	Input/Output Data	0x4000_4000			
SPI0_CONTROL	Module Control	0x4000_4010	Y	Y	
SPI0_CONFIG	Module Configuration	0x4000_4020	Y	Y	
SPI0_CLKRATE	Module Clock Rate Control	0x4000_4030			
SPI0_FSTATUS	FIFO Status	0x4000_4040			
SPI0_CONFIGMD	Mode Configuration	0x4000_4050	Y	Y	
SPI1 Registers					
SPI1_DATA	Input/Output Data	0x4000_5000			
SPI1_CONTROL	Module Control	0x4000_5010	Y	Y	
SPI1_CONFIG	Module Configuration	0x4000_5020	Y	Y	
SPI1_CLKRATE	Module Clock Rate Control	0x4000_5030			
SPI1_FSTATUS	FIFO Status	0x4000_5040			
SPI1_CONFIGMD	Mode Configuration	0x4000_5050	Y	Y	
I2C0 Registers					
I2C0_CONTROL	Module Control	0x4000_9000	Y	Y	
I2C0_CONFIG	Module Configuration	0x4000_9010	Y	Y	
I2C0_SADDRESS	Slave Address	0x4000_9020			
I2C0_SMASK	Slave Address Mask	0x4000_9030			
I2C0_DATA	Data Buffer Access	0x4000_9040			
I2C0_TIMER	Timer Data	0x4000_9050			
I2C0_TIMERRL	Timer Reload Values	0x4000_9060			
I2C0_SCONFIG	SCL Signal Configuration	0x4000_9070			
I2C0_I2CDMA	DMA Configuration	0x4000_9080			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
EPCA0 Registers					
EPCA0_CH0_MODE	Channel Capture/Compare Mode	0x4000_E000			
EPCA0_CH0_CONTROL	Channel Capture/Compare Control	0x4000_E010	Y	Y	
EPCA0_CH0_CCAPV	Channel Compare Value	0x4000_E020			
EPCA0_CH0_CCAPVUPD	Channel Compare Update Value	0x4000_E030			
EPCA0_CH1_MODE	Channel Capture/Compare Mode	0x4000_E040			
EPCA0_CH1_CONTROL	Channel Capture/Compare Control	0x4000_E050	Y	Y	
EPCA0_CH1_CCAPV	Channel Compare Value	0x4000_E060			
EPCA0_CH1_CCAPVUPD	Channel Compare Update Value	0x4000_E070			
EPCA0_CH2_MODE	Channel Capture/Compare Mode	0x4000_E080			
EPCA0_CH2_CONTROL	Channel Capture/Compare Control	0x4000_E090	Y	Y	
EPCA0_CH2_CCAPV	Channel Compare Value	0x4000_E0A0			
EPCA0_CH2_CCAPVUPD	Channel Compare Update Value	0x4000_E0B0			
EPCA0_CH3_MODE	Channel Capture/Compare Mode	0x4000_E0C0			
EPCA0_CH3_CONTROL	Channel Capture/Compare Control	0x4000_E0D0	Y	Y	
EPCA0_CH3_CCAPV	Channel Compare Value	0x4000_E0E0			
EPCA0_CH3_CCAPVUPD	Channel Compare Update Value	0x4000_E0F0			
EPCA0_CH4_MODE	Channel Capture/Compare Mode	0x4000_E100			
EPCA0_CH4_CONTROL	Channel Capture/Compare Control	0x4000_E110	Y	Y	
EPCA0_CH4_CCAPV	Channel Compare Value	0x4000_E120			
EPCA0_CH4_CCAPVUPD	Channel Compare Update Value	0x4000_E130			
EPCA0_CH5_MODE	Channel Capture/Compare Mode	0x4000_E140			
EPCA0_CH5_CONTROL	Channel Capture/Compare Control	0x4000_E150	Y	Y	
EPCA0_CH5_CCAPV	Channel Compare Value	0x4000_E160			
EPCA0_CH5_CCAPVUPD	Channel Compare Update Value	0x4000_E170			
EPCA0_MODE	Module Operating Mode	0x4000_E180			
EPCA0_CONTROL	Module Control	0x4000_E190	Y	Y	

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
EPCA0_STATUS	Module Status	0x4000_E1A0	Y	Y	
EPCA0_COUNTER	Module Counter/Timer	0x4000_E1B0			
EPCA0_LIMIT	Module Upper Limit	0x4000_E1C0			
EPCA0_LIMITUPD	Module Upper Limit Update Value	0x4000_E1D0			
EPCA0_DTIME	Phase Delay Time	0x4000_E1E0			
EPCA0_DTARGET	DMA Transfer Target	0x4000_E200			
TIMER0 Registers					
TIMER0_CONFIG	High and Low Timer Configuration	0x4001_4000	Y	Y	
TIMER0_CLKDIV	Module Clock Divider Control	0x4001_4010			
TIMER0_COUNT	Timer Value	0x4001_4020			
TIMER0_CAPTURE	Timer Capture/Reload Value	0x4001_4030			
TIMER1 Registers					
TIMER1_CONFIG	High and Low Timer Configuration	0x4001_5000	Y	Y	
TIMER1_CLKDIV	Module Clock Divider Control	0x4001_5010			
TIMER1_COUNT	Timer Value	0x4001_5020			
TIMER1_CAPTURE	Timer Capture/Reload Value	0x4001_5030			
TIMER2 Registers					
TIMER2_CONFIG	High and Low Timer Configuration	0x4001_6000	Y	Y	
TIMER2_CLKDIV	Module Clock Divider Control	0x4001_6010			
TIMER2_COUNT	Timer Value	0x4001_6020			
TIMER2_CAPTURE	Timer Capture/Reload Value	0x4001_6030			
SARADC0 Registers					
SARADC0_CONFIG	Module Configuration	0x4001_A000	Y	Y	
SARADC0_CONTROL	Measurement Control	0x4001_A010	Y	Y	
SARADC0_SQ7654	Channel Sequencer Time Slots 4-7 Setup	0x4001_A020			
SARADC0_SQ3210	Channel Sequencer Time Slots 0-3 Setup	0x4001_A030			
SARADC0_CHAR32	Conversion Characteristic 2 and 3 Setup	0x4001_A040	Y	Y	

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
SARADC0_CHAR10	Conversion Characteristic 0 and 1 Setup	0x4001_A050	Y	Y	
SARADC0_DATA	Output Data Word	0x4001_A060			
SARADC0_WCLIMITS	Window Comparator Limits	0x4001_A070			
SARADC0_ACC	Accumulator Initial Value	0x4001_A080			
SARADC0_STATUS	Module Status	0x4001_A090	Y	Y	
SARADC0_FIFOSTATUS	FIFO Status	0x4001_A0A0			
CMP0 Registers					
CMP0_CONTROL	Module Control	0x4001_F000	Y	Y	
CMP0_MODE	Input and Module Mode	0x4001_F010	Y	Y	
CMP1 Registers					
CMP1_CONTROL	Module Control	0x4002_0000	Y	Y	
CMP1_MODE	Input and Module Mode	0x4002_0010	Y	Y	
AES0 Registers					
AES0_CONTROL	Module Control	0x4002_7000	Y	Y	
AES0_XFRSIZE	Number of Blocks	0x4002_7010			
AES0_DATAFIFO	Input/Output Data FIFO Access	0x4002_7020			
AES0_XORFIFO	XOR Data FIFO Access	0x4002_7030			
AES0_HWKEY0	Hardware Key Word 0	0x4002_7040			
AES0_HWKEY1	Hardware Key Word 1	0x4002_7050			
AES0_HWKEY2	Hardware Key Word 2	0x4002_7060			
AES0_HWKEY3	Hardware Key Word 3	0x4002_7070			
AES0_HWKEY4	Hardware Key Word 4	0x4002_7080			
AES0_HWKEY5	Hardware Key Word 5	0x4002_7090			
AES0_HWKEY6	Hardware Key Word 6	0x4002_70A0			
AES0_HWKEY7	Hardware Key Word 7	0x4002_70B0			
AES0_HWCTR0	Hardware Counter Word 0	0x4002_70C0			
AES0_HWCTR1	Hardware Counter Word 1	0x4002_70D0			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
AES0_HWCTR2	Hardware Counter Word 2	0x4002_70E0			
AES0_HWCTR3	Hardware Counter Word 3	0x4002_70F0			
AES0_STATUS	Module Status	0x4002_7100	Y	Y	
ECRC0 Registers					
ECRC0_CONTROL	Module Control	0x4002_8000	Y	Y	
ECRC0_POLY	16-bit Programmable Polynomial	0x4002_8010			
ECRC0_DATA	Input/Result Data	0x4002_8020			
ECRC0_RDATA	Bit-Reversed Output Data	0x4002_8030			
ECRC0_BRDATA	Byte-Reversed Output Data	0x4002_8040			
ECRC0_SCONTROL	Bus Snooping Control	0x4002_8050	Y	Y	
RTC0 Registers					
RTC0_CONFIG	RTC Configuration	0x4002_9000	Y	Y	
RTC0_CONTROL	RTC Control	0x4002_9010	Y	Y	
RTC0_ALARM0	RTC Alarm 0	0x4002_9020			
RTC0_ALARM1	RTC Alarm 1	0x4002_9030			
RTC0_ALARM2	RTC Alarm 2	0x4002_9040			
RTC0_SETCAP	RTC Timer Set/Capture Value	0x4002_9050			
RTC0_LFOCONTROL	LFOSC Control	0x4002_9060			
PBCFG0 Registers					
PBCFG0_CONTROL0	Global Port Control 0	0x4002_A000	Y	Y	
PBCFG0_CONTROL1	Global Port Control 1	0x4002_A010	Y	Y	
PBCFG0_XBAR0	Crossbar 0 Control	0x4002_A020	Y	Y	
PBCFG0_PBKEY	Global Port Key	0x4002_A030			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
PBSTD0 Registers					
PBSTD0_PB	Output Latch	0x4002_A0A0	Y	Y	Y
PBSTD0_PBPIN	Pin Value	0x4002_A0B0			
PBSTD0_PBMDSEL	Mode Select	0x4002_A0C0	Y	Y	
PBSTD0_PBSKIPEN	Crossbar Pin Skip Enable	0x4002_A0D0	Y	Y	
PBSTD0_PBOUTMD	Output Mode	0x4002_A0E0	Y	Y	
PBSTD0_PBDRV	Drive Strength	0x4002_A0F0	Y	Y	
PBSTD0_PM	Port Match Value	0x4002_A100	Y	Y	
PBSTD0_PMEN	Port Match Enable	0x4002_A110	Y	Y	
PBSTD0_PBPGEN	Pulse Generator Pin Enable	0x4002_A120			
PBSTD0_PBPGPHASE	Pulse Generator Phase	0x4002_A130			
PBSTD1 Registers					
PBSTD1_PB	Output Latch	0x4002_A140	Y	Y	Y
PBSTD1_PBPIN	Pin Value	0x4002_A150			
PBSTD1_PBMDSEL	Mode Select	0x4002_A160	Y	Y	
PBSTD1_PBSKIPEN	Crossbar Pin Skip Enable	0x4002_A170	Y	Y	
PBSTD1_PBOUTMD	Output Mode	0x4002_A180	Y	Y	
PBSTD1_PBDRV	Drive Strength	0x4002_A190	Y	Y	
PBSTD1_PM	Port Match Value	0x4002_A1A0	Y	Y	
PBSTD1_PMEN	Port Match Enable	0x4002_A1B0	Y	Y	
PBSTD2 Registers					
PBSTD2_PB	Output Latch	0x4002_A1E0	Y	Y	Y
PBSTD2_PBPIN	Pin Value	0x4002_A1F0			
PBSTD2_PBMDSEL	Mode Select	0x4002_A200	Y	Y	
PBSTD2_PBSKIPEN	Crossbar Pin Skip Enable	0x4002_A210	Y	Y	
PBSTD2_PBOUTMD	Output Mode	0x4002_A220	Y	Y	
PBSTD2_PBDRV	Drive Strength	0x4002_A230	Y	Y	

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
PBSTD2_PM	Port Match Value	0x4002_A240	Y	Y	
PBSTD2_PMEN	Port Match Enable	0x4002_A250	Y	Y	
PBSTD3 Registers					
PBSTD3_PB	Output Latch	0x4002_A280	Y	Y	Y
PBSTD3_PBPIN	Pin Value	0x4002_A290			
PBSTD3_PBMDSEL	Mode Select	0x4002_A2A0	Y	Y	
PBSTD3_PBSKIPEN	Crossbar Pin Skip Enable	0x4002_A2B0	Y	Y	
PBSTD3_PBOUTMD	Output Mode	0x4002_A2C0	Y	Y	
PBSTD3_PBDRV	Drive Strength	0x4002_A2D0	Y	Y	
PBSTD3_PM	Port Match Value	0x4002_A2E0	Y	Y	
PBSTD3_PMEN	Port Match Enable	0x4002_A2F0	Y	Y	
PBGP4 Registers					
PBGP4_PB	Output Latch	0x4002_A320	Y	Y	Y
PBGP4_PBPIN	Pin Value	0x4002_A330			
PBGP4_PBMDSEL	Mode Select	0x4002_A340	Y	Y	
PBGP4_PBOUTMD	Output Mode	0x4002_A350	Y	Y	
PBGP4_PBDRV	Drive Strength	0x4002_A360	Y	Y	
PBGP4_PM	Port Match Value	0x4002_A370	Y	Y	
PBGP4_PMEN	Port Match Enable	0x4002_A380	Y	Y	
RSTSRC0 Registers					
RSTSRC0_RESETEN	System Reset Source Enable	0x4002_C000	Y	Y	
RSTSRC0_RESETFLAG	System Reset Flags	0x4002_C010			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
CLKCTRL0 Registers					
CLKCTRL0_CONTROL	Module Control	0x4002_D000			
CLKCTRL0_AHBCLKG	AHB Clock Gate	0x4002_D010	Y	Y	
CLKCTRL0_APBCLKG0	APB Clock Gate 0	0x4002_D020	Y	Y	
CLKCTRL0_APBCLKG1	APB Clock Gate 1	0x4002_D030	Y	Y	
CLKCTRL0_PM3CN	Power Mode 3 Clock Control	0x4002_D040			
CLKCTRL0_CONFIG	Configuration Options	0x4002_D060	Y	Y	
FLASHCTRL0 Registers					
FLASHCTRL0_CONFIG	Controller Configuration	0x4002_E000	Y	Y	
FLASHCTRL0_WRADDR	Flash Write Address	0x4002_E0A0			
FLASHCTRL0_WRDATA	Flash Write Data	0x4002_E0B0			
FLASHCTRL0_KEY	Flash Modification Key	0x4002_E0C0			
FLASHCTRL0_TCONTROL	Flash Timing Control	0x4002_E0D0			
VMON0 Registers					
VMON0_CONTROL	Module Control	0x4002_F000	Y	Y	
WDTIMER0 Registers					
WDTIMER0_CONTROL	Module Control	0x4003_0000	Y	Y	
WDTIMER0_STATUS	Module Status	0x4003_0010	Y	Y	
WDTIMER0_THRESHOLD	Threshold Values	0x4003_0020			
WDTIMER0_WDTKEY	Module Key	0x4003_0030			
IDAC0 Registers					
IDAC0_CONTROL	Module Control	0x4003_1000	Y	Y	
IDAC0_DATA	Output Data	0x4003_1010			
IDAC0_BUFSTATUS	FIFO Buffer Status	0x4003_1020	Y	Y	
IDAC0_BUFFER10	FIFO Buffer Entries 0 and 1	0x4003_1030			
IDAC0_BUFFER32	FIFO Buffer Entries 2 and 3	0x4003_1040			
IDAC0_GAINADJ	Output Current Gain Adjust	0x4003_1050			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
DMACTRL0 Registers					
DMACTRL0_STATUS	Controller Status	0x4003_6000			
DMACTRL0_CONFIG	Controller Configuration	0x4003_6004			
DMACTRL0_BASEPTR	Base Pointer	0x4003_6008			
DMACTRL0_ABASEPTR	Alternate Base Pointer	0x4003_600C			
DMACTRL0_CHSTATUS	Channel Status	0x4003_6010			
DMACTRL0_CHSWRCN	Channel Software Request Control	0x4003_6014			
DMACTRL0_CHREQMSET	Channel Request Mask Set	0x4003_6020			
DMACTRL0_CHREQMCLR	Channel Request Mask Clear	0x4003_6024			
DMACTRL0_CHENSET	Channel Enable Set	0x4003_6028			
DMACTRL0_CHENCLR	Channel Enable Clear	0x4003_602C			
DMACTRL0_CHALTSET	Channel Alternate Select Set	0x4003_6030			
DMACTRL0_CHALTCLR	Channel Alternate Select Clear	0x4003_6034			
DMACTRL0_CHHPSET	Channel High Priority Set	0x4003_6038			
DMACTRL0_CHHPCLR	Channel High Priority Clear	0x4003_603C			
DMACTRL0_BERRCLR	Bus Error Clear	0x4003_604C			
DMAXBAR0 Registers					
DMAXBAR0_DMAXBAR0	Channel 0-7 Trigger Select	0x4003_7000	Y	Y	
DMAXBAR0_DMAXBAR1	Channel 8-15 Trigger Select	0x4003_7010	Y	Y	
LPTIMER0 Registers					
LPTIMER0_CONTROL	Module Control	0x4003_8000	Y	Y	
LPTIMER0_COUNT	Timer Value	0x4003_8010			
LPTIMER0_THRESHOLD	Threshold Values	0x4003_8020			
LPTIMER0_STATUS	Module Status	0x4003_8030	Y	Y	
LDO0 Registers					
LDO0_CONTROL	Control	0x4003_9000	Y	Y	

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
VREF0 Registers					
VREF0_CONTROL	Module Control	0x4003_9010	Y	Y	
PLL0 Registers					
PLL0_DIVIDER	Reference Divider Setting	0x4003_B000			
PLL0_CONTROL	Module Control	0x4003_B010	Y	Y	
PLL0_SSPR	Spectrum Spreading Control	0x4003_B020			
PLL0_CALCONFIG	Calibration Configuration	0x4003_B030			
EXTOSC0 Registers					
EXTOSC0_CONTROL	Oscillator Control	0x4003_C000	Y	Y	
PVTOSC0 Registers					
PVTOSC0_CONTROL	Module Control	0x4003_D000	Y	Y	
ACCTR0 Registers					
ACCTR0_CONFIG	Configuration	0x4004_2000			
ACCTR0_CONTROL	Control Register	0x4004_2010			
ACCTR0_LCCONFIG	LC Configuration	0x4004_2020			
ACCTR0_TIMING	Timing	0x4004_2030			
ACCTR0_LCMODE	LC Mode	0x4004_2040			
ACCTR0_LCCLKCONTROL	LC Clock Control	0x4004_2050			
ACCTR0_LCLIMITS	LC Counter Limits	0x4004_2060			
ACCTR0_LCCOUNT	LC Counters	0x4004_2070			
ACCTR0_DBCONFIG	Pulse Counter Debounce Configuration	0x4004_2080			
ACCTR0_COUNT0	Pulse Counter 0	0x4004_2090			
ACCTR0_COUNT1	Pulse Counter 1	0x4004_20A0			
ACCTR0_COMP0	Comparator 0	0x4004_20B0			
ACCTR0_COMP1	Pulse Counter Comparator 1 Threshold	0x4004_20C0			
ACCTR0_STATUS	Pulse Counter Status	0x4004_20D0			
ACCTR0_DEBUGEN	Calibration	0x4004_20E0			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
PMU0 Registers					
PMU0_CONTROL	Module Control	0x4004_8000	Y	Y	
PMU0_CONFIG	Module Configuration	0x4004_8010	Y	Y	
PMU0_STATUS	Module Status	0x4004_8020	Y	Y	
PMU0_WAKEEN	Wakeup Enable	0x4004_8030	Y	Y	
PMU0_WAKESTATUS	Wakeup Status	0x4004_8040			
PMU0_PWEN	Pin Wake Pin Enable	0x4004_8050	Y	Y	
PMU0_PWPOL	Pin Wake Pin Polarity Select	0x4004_8060	Y	Y	
LOCK0 Registers					
LOCK0_KEY	Security Key	0x4004_9000			
LOCK0_PERIPHLOCK0	Peripheral Lock Control 0	0x4004_9020	Y	Y	
LOCK0_PERIPHLOCK1	Peripheral Lock Control 1	0x4004_9040	Y	Y	
SCONFIG0 Registers					
SCONFIG0_CONFIG	System Configuration	0x4004_90B0	Y	Y	
DEVICEID0 Registers					
DEVICEID0_DEVICEID0	Device ID Word 0	0x4004_90C0			
DEVICEID0_DEVICEID1	Device ID Word 1	0x4004_90D0			
DEVICEID0_DEVICEID2	Device ID Word 2	0x4004_90E0			
DEVICEID0_DEVICEID3	Device ID Word 3	0x4004_90F0			
DTM0 Registers					
DTM0_CONTROL	Module Control	0x4004_A000	Y	Y	
DTM0_TIMEOUT	Module Timeout	0x4004_A010			
DTM0_MSTCOUNT	Master Counter	0x4004_A020			
DTM0_STATEADDR	State Address	0x4004_A030			
DTM0_STATE	Active DTM State	0x4004_A040			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
DTM1 Registers					
DTM1_CONTROL	Module Control	0x4004_B000	Y	Y	
DTM1_TIMEOUT	Module Timeout	0x4004_B010			
DTM1_MSTCOUNT	Master Counter	0x4004_B020			
DTM1_STATEADDR	State Address	0x4004_B030			
DTM1_STATE	Active DTM State	0x4004_B040			
DTM2 Registers					
DTM2_CONTROL	Module Control	0x4004_C000	Y	Y	
DTM2_TIMEOUT	Module Timeout	0x4004_C010			
DTM2_MSTCOUNT	Master Counter	0x4004_C020			
DTM2_STATEADDR	State Address	0x4004_C030			
DTM2_STATE	Active DTM State	0x4004_C040			
LCD0 Registers					
LCD0_CONFIG	Configuration	0x4004_D000	Y	Y	
LCD0_CLKCONTROL	Clock Control	0x4004_D020			
LCD0_BLKCONTROL	Blinking Control	0x4004_D030			
LCD0_SEGCONTROL	Segment Control	0x4004_D040			
LCD0_CTRSTCONTROL	Contrast Control	0x4004_D060			
LCD0_VBMCONTROL	VBAT Monitor Control	0x4004_D070			
LCD0_SEGMASK0	Segment Mask 0	0x4004_D080	Y	Y	
LCD0_SEGMASK1	Segment Mask 1	0x4004_D090	Y	Y	
LCD0_SEGDATA0	Segment Data 0	0x4004_D0A0			
LCD0_SEGDATA1	Segment Data 1	0x4004_D0B0			
LCD0_SEGDATA2	Segment Data 2	0x4004_D0C0			
LCD0_SEGDATA3	Segment Data 3	0x4004_D0D0			
LCD0_SEGDATA4	Segment Data 4	0x4004_D0E0			

Table 3.1. Register Memory Map

Register Name	Title	Address (ALL Access)	SET (+0x4)	CLR(+0x8)	MSK (+0xC)
DCDC0 Registers					
DCDC0_CONTROL	Module Control	0x4004_E000	Y	Y	
DCDC0_CONFIG	Module Configuration	0x4004_E010			
ENCDEC0 Registers					
ENCDEC0_CONTROL	Module Control	0x4004_F000	Y	Y	
ENCDEC0_STATUS	Module Status	0x4004_F010			
ENCDEC0_DATAIN	Data Input	0x4004_F020			
ENCDEC0_DATAOUT	Data Output	0x4004_F030			
ENCDEC0_DATAOUTC	Data Output Complement	0x4004_F040			

4. Interrupts

SiM3L1xx devices implement the standard nested-vectored interrupt controller (NVIC) available in the ARM Cortex-M3 core. The specific system exceptions, interrupt vectors, and priority implementation are described in the following sections.

4.1. System Exceptions

The system-level exceptions on SiM3L1xx devices are shown in Table 4.1.

Table 4.1. System Exceptions

Exception Number	Type	Priority	Description
1	Reset	–3	System Reset.
2	NMI	–2	External NMI Input.
3	Hard Fault	–1	All fault conditions if the corresponding fault handler is disabled.
4	MemManage Fault	Programmable	MMU/MPU fault (not supported).
5	Bus Fault	Programmable	AHB error received from slave (either prefetch abort or data abort).
6	Usage Fault	Programmable	Exception due to program error.
7	Reserved		
8	Reserved		
9	Reserved		
10	Reserved		
11	SVcall	Programmable	System Service Call using SWI or SVC instruction.
12	Debug Monitor	Programmable	Breakpoint, Watchpoint, or external debug request.
13	Reserved		
14	PendSV	Programmable	Pendable reset for system device.
15	SYSTICK	Programmable	System tick timer.

4.2. Interrupt Vector Table

The interrupt vector table for SiM3L1xx is shown in Table 4.2.

Table 4.2. Interrupt Vector Table

Position	Default Priority	Name/ Description	Sources	Default Address
	–3	Reset	System Reset	0x00000004
	–2	NMI	NMI	0x00000008
	–1	Hard Fault	All fault conditions if the corresponding fault handler is disabled	0x0000000C
	0	MemManage	MMU/MPU fault (not supported)	0x00000010
	1	Bus Fault	AHB error received from slave (either prefetch abort or data abort)	0x00000014
	2	Usage Fault	Exception due to program error	0x00000018
		Reserved		0x0000001C
		Reserved		0x00000020
		Reserved		0x00000024
		Reserved		0x00000028
	3	SVcall	System Service Call using SWI or SVC instruction	0x0000002C
	4	Debug Monitor	Breakpoint Watchpoint External debug request	0x00000030
		Reserved		0x00000034
	5	PendSV	Pendable reset for system device	0x00000038
	6	SYSTICK	System tick timer	0x0000003C
0	7	WDTIMER0	First threshold crossed	0x00000040
1	8	PBEXT0	External pin (INT0.x) rising edge External pin (INT0.x) falling edge	0x00000044
2	9	PBEXT1	External pin (INT1.x) rising edge External pin (INT1.x) falling edge	0x00000048
3	10	RTC0ALRM	Alarm 0 Alarm 1 Alarm 2	0x0000004C
4	11	LPTIMER0	Timer overflow Compare threshold crossed	0x00000050
5	12	DMAERR	DMA error	0x00000054
6	13	DMACH0	DMA Channel 0 done	0x00000058
7	14	DMACH1	DMA Channel 1 done	0x0000005C

Table 4.2. Interrupt Vector Table (Continued)

Position	Default Priority	Name/ Description	Sources	Default Address
8	15	DMACH2	DMA Channel 2 done	0x00000060
9	16	DMACH3	DMA Channel 3 done	0x00000064
10	17	DMACH4	DMA Channel 4 done	0x00000068
11	18	DMACH5	DMA Channel 5 done	0x0000006C
12	19	DMACH6	DMA Channel 6 done	0x00000070
13	20	DMACH7	DMA Channel 7 done	0x00000074
14	21	DMACH8	DMA Channel 8 done	0x00000078
15	22	DMACH9	DMA Channel 9 done	0x0000007C
16	23	TIMER0L	TIMER0 Low overflow	0x00000080
17	24	TIMER0H	TIMER0 High overflow	0x00000084
18	25	TIMER1L	TIMER1 Low overflow	0x00000088
19	26	TIMER1H	TIMER1 High overflow	0x0000008C
20	27	TIMER2L	TIMER2 Low overflow	0x00000090
21	28	TIMER2H	TIMER2 High overflow	0x00000094
22	29	ACCTR0	Flutter start Flutter stop Quadrature encoder Integrator transition Digital comparator 1 Digital comparator 0 Counter overflow Direction change	0x00000098
23	30	EPCA0	Counter overflow Halt external signal is high Channel compare or match Channel intermediate overflow	0x0000009C
24	31	USART0	Receive frame error Receive parity error Receive overrun Receive data request Transmit SmartCard parity error Transmit underrun Transmit data request Transmit complete	0x000000A0

Table 4.2. Interrupt Vector Table (Continued)

Position	Default Priority	Name/ Description	Sources	Default Address
25	32	UART0	Receive frame error Receive parity error Receive overrun Receive data request Transmit data request Transmit complete	0x000000A4
26	33	SPI0	Shift Register empty FIFO underrun Mode fault Slave Select pin Illegal receive FIFO access Receive FIFO Overrun Receive FIFO read request Illegal transmit FIFO access Transmit FIFO overrun Transmit FIFO write request	0x000000A8
27	34	SPI1	Shift Register empty FIFO underrun Mode fault Slave Select pin Illegal receive FIFO access Receive FIFO Overrun Receive FIFO read request Illegal transmit FIFO access Transmit FIFO overrun Transmit FIFO write request	0x000000AC
28	35	I2C0	Start Transmit complete Receive complete Acknowledge Stop Timer byte 0 overflow Timer byte 1 overflow Timer byte 2 overflow Timer byte 3 overflow Arbitration lost	0x000000B0
29	36	SARADC0	Conversion complete Scan complete FIFO underrun FIFO overrun Window comparator threshold crossed	0x000000B4
30	37	CMP0	Rising edge occurred Falling edge occurred	0x000000B8

Table 4.2. Interrupt Vector Table (Continued)

Position	Default Priority	Name/ Description	Sources	Default Address
31	38	CMP1	Rising edge occurred Falling edge occurred	0x000000BC
32	39	DTM0	State transition occurred Transfer complete Timeout error occurred	0x000000C0
33	40	DTM1	State transition occurred Transfer complete Timeout error occurred	0x000000C4
34	41	DTM2	State transition occurred Transfer complete Timeout error occurred	0x000000C8
35	42	AES0	Operation complete Error occurred	0x000000CC
36	43	ENCDEC0	Input ready Output ready Data error Data underrun Data overrun	0x000000D0
37	44	RTC0FAIL	RTC0 Oscillator failed	0x000000D4
38	45	VBATLOW	VBAT falls below the early warning threshold	0x000000D8
39	46	PMU0	Charge pump fail	0x000000DC
40	47	DCDC0	Output low Output good Output not in regulation Output in regulation	0x000000E0
41	48	PMATCH0	Port Match event PMU Pin Wake event	0x000000E4
42	49	IDAC0	Data buffer overrun Data buffer underrun Data buffer went empty	0x000000E8
43	50	PLL0	Lock saturation high or low Oscillator lock	0x000000EC

4.3. Priorities

The SiM3L1xx devices implement 3 bits of interrupt priority, as shown in Figure 4.1.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Programmable Interrupt Priority Level			Reserved				

Figure 4.1. SiM3L1xx Interrupt Priorities

In addition to the different priority levels, the NVIC allows for different priority groups, as shown in Table 4.3. These groups determine the number of bits used to determine the Preempt Priority and Subpriority settings for each interrupt. A higher priority interrupt can preempt or interrupt a lower priority interrupt. If two interrupts of the same priority occur at the same time, the interrupts cannot preempt each other and the interrupt with the higher Subpriority (lowest value) will be taken first. The Reset, NMI, and Hard Fault exceptions have fixed negative priorities to always take precedence over other interrupts in the system.

Table 4.3. Priority Groups

Priority Group	Preempt Priority Field Size	Subpriority Field Size
0–3	[7:4]	None
4	[7:5]	[4]
5	[7:6]	[5:4]
6	[7]	[6:4]
7	None	[7:4]

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Preempt Priority Level			Subpriority	Reserved			

Figure 4.2. Priority Group 4 Fields

Table 4.4. Priority Levels with Priority Group 4

Preempt Priority	Subpriority	Priority Value
1 (Highest)	1 (Highest)	0x00
1	2	0x10
2	1	0x20
2	2	0x30
3	1	0x40
3	2	0x50
4	1	0x60
4	2	0x70
5	1	0x80
5	2	0x90
6	1	0xA0
6	2	0xB0
7	1	0xC0
7	2	0xD0
8	1	0xE0
8 (Lowest)	2 (Lowest)	0xF0

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Subpriority Level				Reserved			

Figure 4.3. Priority Group 7 Fields

Table 4.5. Priority Levels with Priority Group 7 (Interrupts Cannot Preempt)

Preempt Priority	Subpriority	Priority Value
	1 (Highest)	0x00
	2	0x10
	3	0x20
	4	0x30
	5	0x40
	6	0x50
	7	0x60
	8	0x70
	9	0x80
	10	0x90
	11	0xA0
	12	0xB0
	13	0xC0
	14	0xD0
	15	0xE0
	16 (Lowest)	0xF0

The Priority Group and Interrupt Priority settings are in the NVIC.

5. Clock Control (CLKCTRL0)

This section describes the Clock Control (CLKCTRL) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

5.1. Clock Control Features

Clock Control includes the following features:

- Support for multiple oscillator sources for system clock with smooth and glitch-less transition.
- Individual clock gating controls for most peripherals and modules.
- Multiple options for AHB clock settings and a divider for the APB clock.
- Synchronization between AHB and APB clocks.

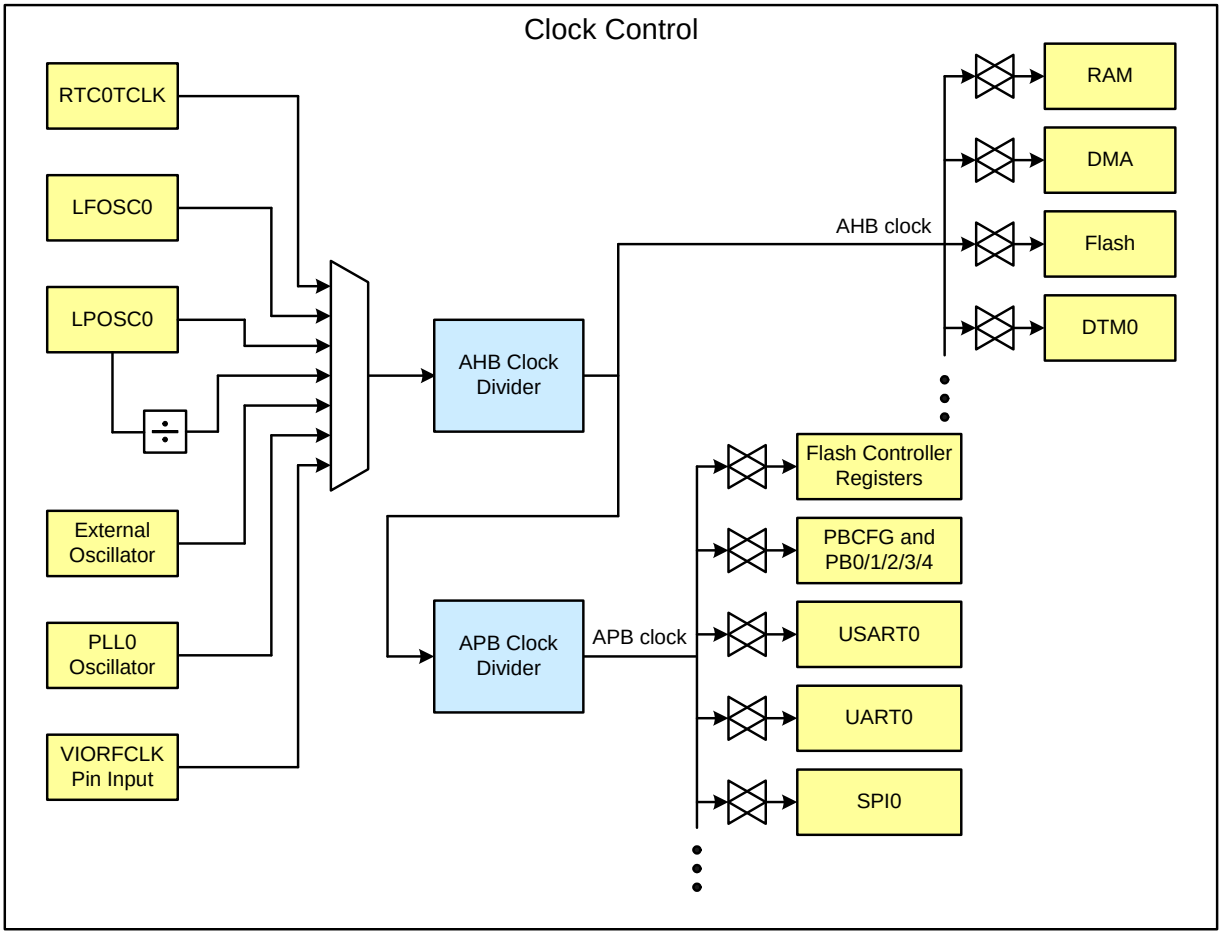


Figure 5.1. Clock Control Block Diagram

Clock Control generates the two system clocks: AHB and APB. The AHB clock services memory peripherals and can be derived from one of seven sources: the RTC0 timer clock, the Low Frequency Oscillator, the Low Power Oscillator, the divided Low Power Oscillator, the External Oscillator, the PLL0 Oscillator, and the VIORFCLK pin input. In addition, a divider for the AHB clock provides flexible clock options for the device. The APB clock services data peripherals and is synchronized with the AHB clock. The APB clock can be equal to the AHB clock or set to the AHB clock divided by two.

Clock Control allows the AHB and APB clocks to be turned off to unused peripherals to save system power. Any registers in a peripheral with disabled clocks will be unable to be accessed (read or write) until the clocks are enabled. Most peripherals have clocks off by default after a power-on reset.

5.2. CLKCTRL0 Registers

This section contains the detailed register descriptions for CLKCTRL0 registers.

Register 5.1. CLKCTRL0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EXTOSCEN	VIORFCLKEN	OBUSYF	EXTESEL	Reserved											APBDIV
Type	RW	RW	R	RW	R											RW
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					AHBDIV			Reserved					AHBSEL		
Type	R					RW			R					RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

CLKCTRL0_CONTROL = 0x4002_D000

Table 5.1. CLKCTRL0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	EXTOSCEN	**External Clock Input Enable.** This bit must be set to 1 before selecting the EXTOSC or EXTOSC/2 as the clock source for the AHB. 0: Disable the EXTOSC input. 1: Enable the EXTOSC input.
30	VIORFCLKEN	**VIORF Clock Enable.** This bit must be set to 1 before selecting the VIORFCLK pin as the clock source for the AHB. 0: Disable the VIORFCLK input. 1: Enable the VIORFCLK input.
29	OBUSYF	**Oscillators Busy Flag.** When set, this status bit indicates that a requested clock switch-over is in progress. Firmware should wait until the flag is clear to reconfigure the AHBSEL, AHBDIV, and APBDIV fields.

Table 5.1. CLKCTRL0_CONTROL Register Bit Descriptions

Bit	Name	Function
28	EXTESEL	**External Clock Edge Select.** Select the edge mode used by the external clock for the TIMER and EPCA modules. This external clock is synchronized with the APB clock. 0: External clock generated by both rising and falling edges of the external oscillator. 1: External clock generated by only rising edges of the external oscillator.
27:17	Reserved	Must write reset value.
16	APBDIV	**APB Clock Divider.** Divides the APB clock from the AHB clock. This field should not be modified when OBUSYF is set. 0: APB clock is the same as the AHB clock (divided by 1). 1: APB clock is the AHB clock divided by 2.
15:11	Reserved	Must write reset value.
10:8	AHBDIV	**AHB Clock Divider.** Divides the AHB clock. This field should not be modified when OBUSYF is set. 000: AHB clock divided by 1. 001: AHB clock divided by 2. 010: AHB clock divided by 4. 011: AHB clock divided by 8. 100: AHB clock divided by 16. 101: AHB clock divided by 32. 110: AHB clock divided by 64. 111: AHB clock divided by 128.
7:3	Reserved	Must write reset value.
2:0	AHBSEL	**AHB Clock Source Select.** This field should not be modified when OBUSYF is set. 000: AHB clock source is the Low-Power Oscillator. 001: AHB clock source is the Low-Frequency Oscillator. 010: AHB clock source is the RTC0TCLK signal. 011: AHB clock source is the External Oscillator. 100: AHB clock source is the VIORFCLK input pin. 101: AHB clock source is the PLL. 110: AHB clock source is a divided version of the Low-Power Oscillator. 111: Reserved.

Register 5.2. CLKCTRL0_AHBCLKG: AHB Clock Gate

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved										DTM2EN	DTM1EN	DTM0EN	FLASHCEN	DMACEN	RAMCEN
Type	R										RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Register ALL Access Address

CLKCTRL0_AHBCLKG = 0x4002_D010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 5.2. CLKCTRL0_AHBCLKG Register Bit Descriptions

Bit	Name	Function
31:6	Reserved	Must write reset value.
5	DTM2EN	**DTM2 Clock Enable.** 0: Disable the AHB clock to Data Transfer Manager 2 (DTM2). 1: Enable the AHB clock to Data Transfer Manager 2 (DTM2).
4	DTM1EN	**DTM1 Clock Enable.** 0: Disable the AHB clock to Data Transfer Manager 1 (DTM1). 1: Enable the AHB clock to Data Transfer Manager 1 (DTM1).
3	DTM0EN	**DTM0 Clock Enable.** 0: Disable the AHB clock to Data Transfer Manager 0 (DTM0). 1: Enable the AHB clock to Data Transfer Manager 0 (DTM0).
2	FLASHCEN	**Flash Clock Enable.** 0: Disable the AHB clock to the Flash. 1: Enable the AHB clock to the Flash.
1	DMACEN	**DMA Clock Enable.** 0: Disable the AHB clock to the DMA Controller. 1: Enable the AHB clock to the DMA Controller.
0	RAMCEN	**RAM Clock Enable.** 0: Disable the AHB clock to the RAM. 1: Enable the AHB clock to the RAM.

Register 5.3. CLKCTRL0_APBCLKG0: APB Clock Gate 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved						PLL0CEN	ENCDEC0CEN	DCDC0CEN	LCD0CEN	DTM2CEN	DTM1CEN	DTM0CEN	ACCTR0CEN	LPT0CEN	IDAC0CEN
Type	R						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CRC0CEN	AES0CEN	CMP1CEN	CMP0CEN	ADC0CEN	TIMER2CEN	TIMER1CEN	TIMER0CEN	EPCA0CEN	I2C0CEN	SPI1CEN	SPI0CEN	UART0CEN	USART0CEN	PB0CEN	FLCTRLCEN
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

CLKCTRL0_APBCLKG0 = 0x4002_D020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 5.3. CLKCTRL0_APBCLKG0 Register Bit Descriptions

Bit	Name	Function
31:26	Reserved	Must write reset value.
25	PLL0CEN	**PLL0 Clock Enable.** 0: Disable the APB clock to the PLL0 registers. 1: Enable the APB clock to the PLL0 registers.
24	ENCDEC0CEN	**ENCDEC0 Clock Enable.** 0: Disable the APB clock to the ENCDEC0 Module. 1: Enable the APB clock to the ENCDEC0 Module.
23	DCDC0CEN	**DCDC0 Clock Enable.** 0: Disable the APB clock to the DCDC0 Module. 1: Enable the APB clock to the DCDC0 Module.
22	LCD0CEN	**LCD0 Clock Enable.** 0: Disable the APB clock to the LCD0 Module. 1: Enable the APB clock to the LCD0 Module.
21	DTM2CEN	**DTM2 Clock Enable.** 0: Disable the APB clock to the DTM2 Register interface. 1: Enable the APB clock to the DTM2 Register interface.

Table 5.3. CLKCTRL0_APBCLKG0 Register Bit Descriptions

Bit	Name	Function
20	DTM1CEN	**DTM1 Clock Enable.** 0: Disable the APB clock to the DTM1 Register interface. 1: Enable the APB clock to the DTM1 Register interface.
19	DTM0CEN	**DTM0 Clock Enable.** 0: Disable the APB clock to the DTM0 Register interface. 1: Enable the APB clock to the DTM0 Register interface.
18	ACCTR0CEN	**ACCTR0 Enable.** 0: Disable the APB clock to the ACCTR0 Module. 1: Enable the APB clock to the ACCTR0 Module.
17	LPT0CEN	**LPT0 Clock Enable.** 0: Disable the APB clock to the LPTIMER0 Module. 1: Enable the APB clock to the LPTIMER0 Module.
16	IDAC0CEN	**IDAC0 Clock Enable.** 0: Disable the APB clock to the IDAC0 Module. 1: Enable the APB clock to the IDAC0 Module.
15	CRC0CEN	**CRC0 Clock Enable.** 0: Disable the APB clock to the CRC0 Module. 1: Enable the APB clock to the CRC0 Module.
14	AES0CEN	**AES0 Clock Enable.** 0: Disable the APB clock to the AES0 Module. 1: Enable the APB clock to the AES0 Module.
13	CMP1CEN	**CMP1 Clock Enable.** 0: Disable the APB clock to the Comparator 1 Module. 1: Enable the APB clock to the Comparator 1 Module.
12	CMP0CEN	**CMP0 Clock Enable.** 0: Disable the APB clock to the Comparator 0 Module. 1: Enable the APB clock to the Comparator 0 Module.
11	ADC0CEN	**SARADC0 Clock Enable.** 0: Disable the APB clock to the SARADC0 Module. 1: Enable the APB clock to the SARADC0 Module.
10	TIMER2CEN	**TIMER2 Clock Enable.** 0: Disable the APB clock to the TIMER2 Module. 1: Enable the APB clock to the TIMER2 Module.
9	TIMER1CEN	**TIMER1 Clock Enable.** 0: Disable the APB clock to the TIMER1 Module. 1: Enable the APB clock to the TIMER1 Module.
8	TIMER0CEN	**TIMER0 Clock Enable.** 0: Disable the APB clock to the TIMER0 Module. 1: Enable the APB clock to the TIMER0 Module.

Table 5.3. CLKCTRL0_APBCLKG0 Register Bit Descriptions

Bit	Name	Function
7	EPCA0CEN	**EPCA0 Clock Enable.** 0: Disable the APB clock to the EPCA0 Module. 1: Enable the APB clock to the EPCA0 Module.
6	I2C0CEN	**I2C0 Clock Enable.** 0: Disable the APB clock to the I2C0 Module. 1: Enable the APB clock to the I2C0 Module.
5	SPI1CEN	**SPI1 Clock Enable.** 0: Disable the APB clock to the SPI1 Module. 1: Enable the APB clock to the SPI1 Module.
4	SPI0CEN	**SPI0 Clock Enable.** 0: Disable the APB clock to the SPI0 Module. 1: Enable the APB clock to the SPI0 Module.
3	UART0CEN	**UART0 Clock Enable.** 0: Disable the APB clock to the UART0 Module. 1: Enable the APB clock to the UART0 Module.
2	USART0CEN	**USART0 Clock Enable.** 0: Disable the APB clock to the USART0 Module. 1: Enable the APB clock to the USART0 Module.
1	PB0CEN	**Port Bank Clock Enable.** 0: Disable the APB clock to the Port Bank Modules. 1: Enable the APB clock to the Port Bank Modules.
0	FLCTRLCEN	**Flash Controller Clock Enable.** 0: Disable the APB clock to the Flash Controller Module (FLASHCTRL0). 1: Enable the APB clock to the Flash Controller Module (FLASHCTRL0).

Register 5.4. CLKCTRL0_APBCLKG1: APB Clock Gate 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved														MISC1CEN	MISC0CEN
Type	R													RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Register ALL Access Address

CLKCTRL0_APBCLKG1 = 0x4002_D030

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 5.4. CLKCTRL0_APBCLKG1 Register Bit Descriptions

Bit	Name	Function
31:2	Reserved	Must write reset value.
1	MISC1CEN	**Miscellaneous 1 Clock Enable.** 0: Disable the APB clock to the Watchdog Timer (WDTIMER0) and DMA Crossbar (DMAXBAR0) modules. 1: Enable the APB clock to the Watchdog Timer (WDTIMER0) and DMA Crossbar (DMAXBAR0) modules.
0	MISC0CEN	**Miscellaneous 0 Clock Enable.** 0: Disable the APB clock to the VMON0, LDO0, EXTOSC0, LPOSC0, RTC0 and RSTSRC modules. 1: Enable the APB clock to the VMON0, LDO0, EXTOSC0, LPOSC0, RTC0 and RSTSRC modules.

Register 5.5. CLKCTRL0_PM3CN: Power Mode 3 Clock Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															PM3CEN
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved													PM3CSEL		
Type	R													RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

CLKCTRL0_PM3CN = 0x4002_D040

Table 5.5. CLKCTRL0_PM3CN Register Bit Descriptions

Bit	Name	Function
31:17	Reserved	Must write reset value.
16	PM3CEN	**Power Mode 3 Fast-Wake Clock Enable.** When set to 1, the core will automatically switch to the clock source defined by PM3CSEL during Power Mode 3, which speeds up the wakeup time. 0: Disable the core clock when in Power Mode 3. 1: The core clock is enabled and runs off the clock selected by PM3CSEL in Power Mode 3.
15:3	Reserved	Must write reset value.
2:0	PM3CSEL	**Power Mode 3 Fast-Wake Clock Source.** If PM3CEN is set to 1, this clock selection should be the LFOSC0 or RTC0OSC clock to save power while in Power Mode 3. Additionally, the AHB and APB source must be set to the Low-Power Oscillator or divided version of the Low-Power Oscillator. 000: Power Mode 3 clock source is the Low-Power Oscillator. 001: Power Mode 3 clock source is the Low-Frequency Oscillator. 010: Power Mode 3 clock source is the RTC0TCLK signal. 011: Power Mode 3 clock source is the External Oscillator. 100: Power Mode 3 clock source is the VIORFCLK input pin. 101: Power Mode 3 clock source is the PLL. 110: Power Mode 3 clock source is a divided version of the Low-Power Oscillator. 111: Reserved.

Register 5.6. CLKCTRL0_CONFIG: Configuration Options

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															PMSEL
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

CLKCTRL0_CONFIG = 0x4002_D060

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 5.6. CLKCTRL0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:1	Reserved	Must write reset value.
0	PMSEL	**Power Mode Select.** This bit is used to enable the reset circuitry to wake the device from power mode 8 (PM8). It should be set to 1 before entering PM8. This bit should be cleared to 0 for all other low power modes.

5.3. CLKCTRL0 Register Memory Map

Table 5.7. CLKCTRL0 Memory Map

Register Name	CLKCTRL0_APBCLKG0	CLKCTRL0_AHBCLKG	CLKCTRL0_CONTROL
ALL Address	0x4002_D020	0x4002_D010	0x4002_D000
Access Methods	ALL | SET | CLR	ALL | SET | CLR	ALL
Bit 31	Reserved	Reserved	EXTOSCEN
Bit 30			VIORFCLKEN
Bit 29			OBUSYF
Bit 28			EXTESEL
Bit 27			Reserved
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			APBDIV
Bit 15			Reserved
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			AHBDIV
Bit 9			
Bit 8			
Bit 7			Reserved
Bit 6			
Bit 5		DTM2EN	
Bit 4		DTM1EN	
Bit 3		DTM0EN	
Bit 2		FLASHCEN	AHBSEL
Bit 1		DMACEN	
Bit 0		RAMCEN	
	PLL0CEN		
	ENCDEC0CEN		
	DCDC0CEN		
	LCD0CEN		
	DTM2CEN		
	DTM1CEN		
	DTM0CEN		
	ACCTR0CEN		
	LPT0CEN		
	IDAC0CEN		
	CRC0CEN		
	AES0CEN		
	CMP1CEN		
	CMP0CEN		
	ADC0CEN		
	TIMER2CEN		
	TIMER1CEN		
	TIMER0CEN		
	EPCA0CEN		
	I2C0CEN		
	SPI1CEN		
	SPI0CEN		
	UART0CEN		
	USART0CEN		
	PB0CEN		
	FLCTRLCEN		

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 5.7. CLKCTRL0 Memory Map

Register Name	CLKCTRL0_APBCLKG1																														
ALL Address	0x4002_D030																														
Access Methods	ALL | SET | CLR																														
Bit 31	Reserved																														
Bit 30																															
Bit 29																															
Bit 28																															
Bit 27																															
Bit 26																															
Bit 25																															
Bit 24																															
Bit 23																															
Bit 22																															
Bit 21																															
Bit 20																															
Bit 19																															
Bit 18																															
Bit 17																															
Bit 16																															
Bit 15																															
Bit 14																															
Bit 13																															
Bit 12																															
Bit 11																															
Bit 10																															
Bit 9																															
Bit 8																															
Bit 7																															
Bit 6																															
Bit 5																															
Bit 4																															
Bit 3																															
Bit 2																															
Bit 1	MISC1CEN																														
Bit 0	MISC0CEN																														

Register Name	CLKCTRL0_PM3CN																														
ALL Address	0x4002_D040																														
Access Methods	ALL																														
Bit 31	Reserved																														
Bit 30																															
Bit 29																															
Bit 28																															
Bit 27																															
Bit 26																															
Bit 25																															
Bit 24																															
Bit 23																															
Bit 22																															
Bit 21																															
Bit 20																															
Bit 19																															
Bit 18																															
Bit 17																															
Bit 16	PM3CEN																														
Bit 15	Reserved																														
Bit 14																															
Bit 13																															
Bit 12																															
Bit 11																															
Bit 10																															
Bit 9																															
Bit 8																															
Bit 7																															
Bit 6																															
Bit 5																															
Bit 4																															
Bit 3																															
Bit 2	PM3CSEL																														
Bit 1																															
Bit 0																															

Register Name	CLKCTRL0_CONFIG																														
ALL Address	0x4002_D060																														
Access Methods	ALL | SET | CLR																														
Bit 31	Reserved																														
Bit 30																															
Bit 29																															
Bit 28																															
Bit 27																															
Bit 26																															
Bit 25																															
Bit 24																															
Bit 23																															
Bit 22																															
Bit 21																															
Bit 20																															
Bit 19																															
Bit 18																															
Bit 17																															
Bit 16																															
Bit 15																															
Bit 14																															
Bit 13																															
Bit 12																															
Bit 11																															
Bit 10																															
Bit 9																															
Bit 8																															
Bit 7																															
Bit 6																															
Bit 5																															
Bit 4																															
Bit 3																															
Bit 2																															
Bit 1																															
Bit 0	PMSEL																														

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

6. Reset Sources (RSTSRC0)

This section describes the Reset Sources (RSTSRC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

6.1. Reset Sources Features

The Reset Source block includes the following features:

- Separate enable mask bits for each reset source other than the $\overline{\text{RESET}}$ pin.
- Separate flags for each reset source indicating the cause of the last reset.

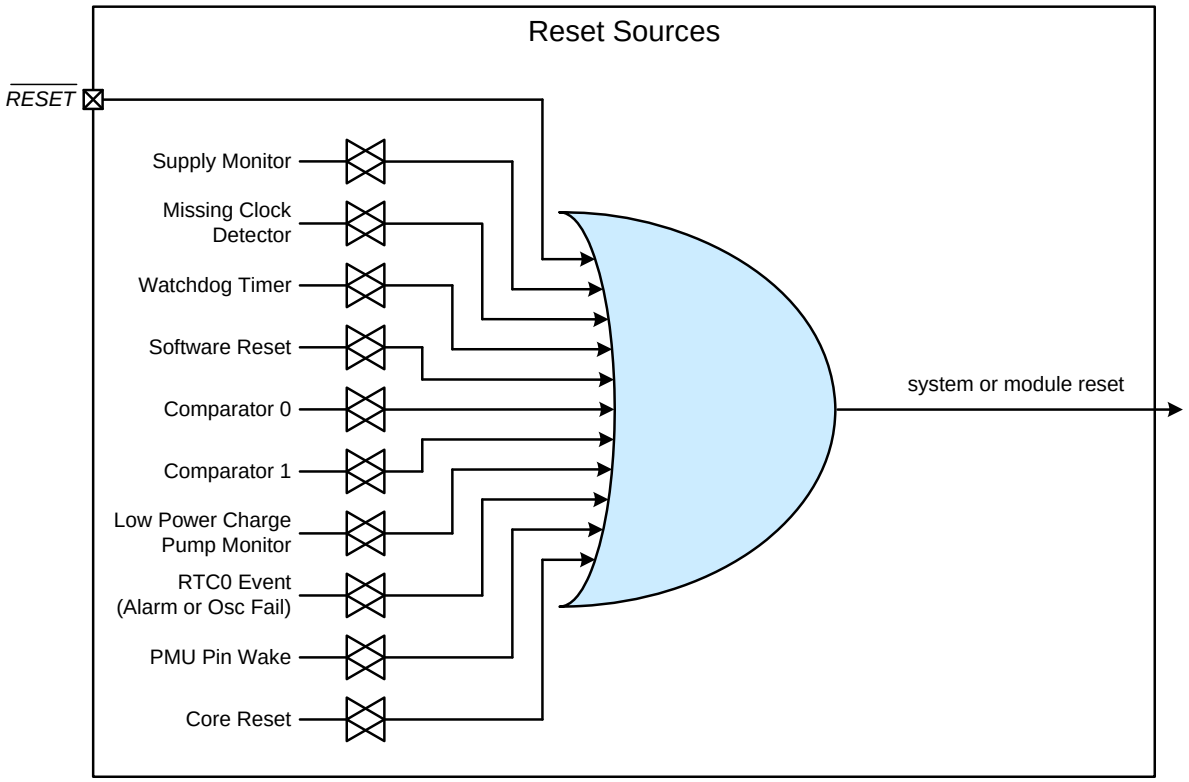


Figure 6.1. Reset Sources Block Diagram

Reset circuitry allows the controller to be easily placed in a predefined default condition. On entry to this reset state, the following occur:

- The core halts program execution.
- Module registers are initialized to their defined reset values unless the bits reset only with a power-on reset.
- External port pins are forced to a known state.
- Interrupts and timers are disabled.
- AHB peripherals clocks to flash and RAM are enabled.
- Clocks to all APB peripherals other than Watchdog Timer, EMIF0, and DMAXBAR are disabled.

All registers are reset to the predefined values noted in the register descriptions unless the bits only reset with a power-on reset. The contents of RAM are unaffected during a reset; any previously stored data is preserved as long as power is not lost.

The Port I/O latches are reset to 1 in open-drain mode. Weak pullups are enabled during and after the reset. For VBAT Supply Monitor and power-on resets, the $\overline{\text{RESET}}$ pin is driven low until the device exits the reset state.

On exit from the reset state, the program counter (PC) is reset, and the system clock defaults to the internal low-power oscillator. The Watchdog Timer is enabled with the low frequency oscillator as its clock source. Program execution begins at location 0x00000000.

6.1.1. Power-On Reset

During power-up, the device is held in a reset state and the $\overline{\text{RESET}}$ pin voltage is held low until the device is released from reset. After VBAT settles above V_{POR}, a delay occurs before the device is released from reset; the delay decreases as the VBAT ramp time increases (VBAT ramp time is defined as how fast VBAT ramps from 0 V to V_{POR}). Figure 6.2 plots the power-on and VBAT monitor reset timing. For valid ramp times, the power-on reset delay (t_{POR}) is given in the electrical specifications chapter of the device data sheet.

Note: VBAT ramp times slower than the maximum may cause the device to be released from reset before VBAT reaches the V_{POR} level.

On exit from a power-on reset, the PORRF flag is set by hardware. When PORRF or VMONRF is set, all of the other reset flags in the RESETFLAG Register are indeterminate. Since all resets cause program execution to begin at the same location, firmware can read the PORF flag in the PMU STATUS register to determine if a power-up was the cause of reset. The contents of internal data memory, including retention RAM should be assumed to be undefined after a power-on reset.

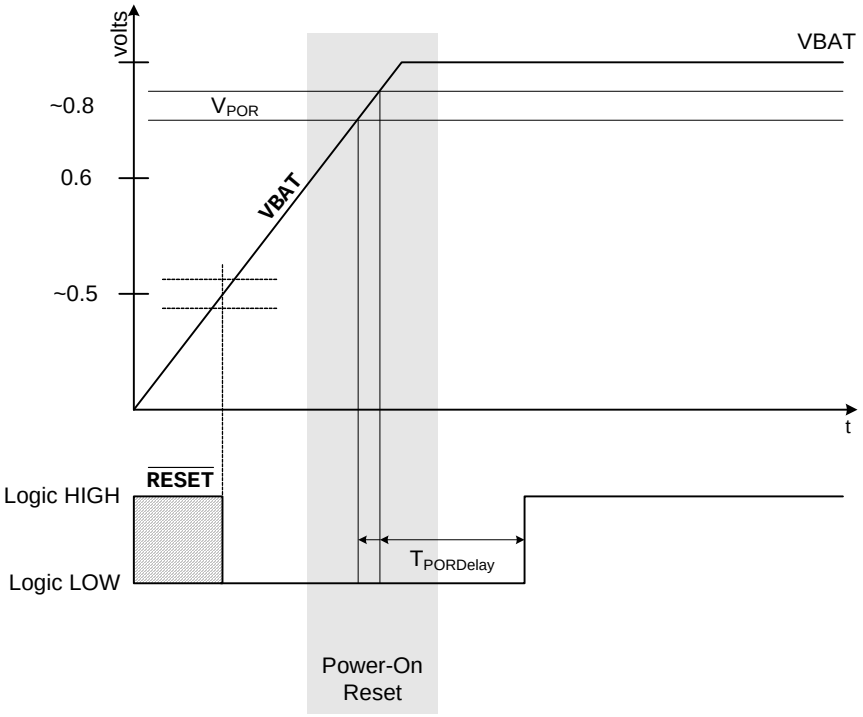


Figure 6.2. Power-On Reset Timing Diagram

6.1.2. VBAT Monitor Power-Fail Reset

SiM3L1xx devices have a VBAT Supply Monitor that is enabled and selected as a reset source after each power-on. When enabled and selected as a reset source, any power down transition or power irregularity that causes VBAT to drop below VRST will cause the RESET pin to be driven low and the core will be held in a reset state. When VBAT returns to a level above VRST, the core will be released from the reset state.

After a power-fail reset, the VBATMRF flag reads 1, all of the other reset flags in the RESETFLAG Register are indeterminate, the contents of RAM are invalid, and the VBAT Monitor is enabled and selected as a reset source. The enable state of the VBAT Monitor and its selection as a reset source is only altered by power-on and power-fail resets. For example, if the VBAT supply monitor is de-selected as a reset source and disabled by firmware, then a software reset is performed, the VBAT Monitor will remain disabled and de-selected after the reset.

To provide firmware early notification that a power failure is about to occur, the VBATHI bit is cleared when the VBAT supply falls below the VBAT High threshold. The VBATHI bit can be configured to generate an interrupt.

Note: To protect the integrity of Flash contents, the VBAT Monitor must be enabled and selected as a reset source if firmware contains routines which erase or write Flash memory. If the VBAT Monitor is not enabled, any erase or write performed on Flash memory will be ignored.

Important Notes:

- The Power-on Reset (POR) delay is not incurred after a VBAT supply monitor reset.
- Firmware should take care not to inadvertently disable the VBAT Monitor as a reset source when writing to RESETEN to enable other reset sources or to trigger a software reset. All writes to RESETEN should explicitly set VBATMREN to 1 to keep the VBAT Monitor enabled as a reset source.
- The VBAT Monitor must be enabled before selecting it as a reset source. Selecting the VBAT Monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the VBAT Monitor and selecting it as a reset source. The procedure for enabling the VBAT Monitor and selecting it as a reset source from a disabled state is:

1. Enable the VBAT Monitor.
2. Wait for the VBAT Monitor to stabilize (optional).
3. Select the VBAT Monitor as a reset source (VBATMREN bit).

6.1.3. External Reset

The external RESET pin provides a means for external circuitry to force the device into a reset state. Asserting an active-low signal on the RESET pin generates a reset; an external pull-up and/or decoupling of the RESET pin may be necessary to avoid erroneous noise-induced resets. It is not recommended to tie RESET directly to a supply pin. The external reset remains functional even when the device is in the low power modes. The PINRF flag is set on exit from an external reset.

6.1.4. Missing Clock Detector Reset

The missing clock detector (MCD) is a one-shot circuit that is triggered by the APB clock. The APB clock is derived from the AHB clock, so monitoring the APB clock will detect a failure in either clock tree. If the APB clock remains high or low for longer than the Missing Clock Detector Timeout, the one-shot will time out and generate a reset. After a MCD reset, the MCDRF flag will read 1, signifying the MCD as the reset source; otherwise, this bit reads 0. Writing a 1 to the MCDREN bit enables the Missing Clock Detector; writing a 0 disables it. The missing clock detector reset is automatically disabled when the device is in the low power modes. Upon exit from either low power state, the enabled/disabled state of this reset source is restored to its previous value. The state of the RESET pin is unaffected by this reset. Missing clock detector timeout values are given in the device data sheet electrical specifications tables.

6.1.5. Comparator Reset

Comparator 0 (CMP0) or Comparator 1 (CMP1) can be configured as a reset source by writing a 1 to the CMP0REN or CMP1REN bit. The Comparator should be enabled and allowed to settle prior to writing to the enable bits to prevent any turn-on chatter on the output from generating an unwanted reset. The Comparator reset is active-low: if the non-inverting input voltage (on CP0+) is less than the inverting input voltage (on CP0–), the device

is put into the reset state. After a Comparator reset, the CMP0RF or CMP1RF flag will read 1 signifying Comparator 0 or Comparator 1 as the reset source; otherwise, these bits read 0. The Comparator 0 reset source remains functional even when the device is in the low power modes as long as Comparator 0 is also enabled as a wake-up source. The state of the RESET pin is unaffected by this reset.

6.1.6. Watchdog Timer Reset

The Watchdog Timer (WDTIMER0) can be used to recover from certain types of system malfunctions. If a system malfunction prevents user firmware from updating the Watchdog Timer, a reset is generated and the WDTRF bit is set to 1. The Watchdog Timer can be enabled or disabled as a reset source using the WDTREN bit. Note that WDTREN will always read back 1 if the Watchdog Timer is ever disabled. The Watchdog Timer is automatically disabled as a reset source when the device is in the low power modes. Upon exit from either low power state, the enabled/disabled state of this reset source is restored to its previous value.The state of the RESET pin is unaffected by this reset.

6.1.7. RTC Reset

The RTC0 Module can generate a system reset on two events: RTC0 Oscillator Fail or RTC0 Alarm 0. The RTC0 Oscillator Fail event occurs when the RTC0 Missing Clock Detector is enabled and the RTC0OSC falls below the missing clock detector trigger frequency. An RTC0 Alarm event occurs when the RTC0 Alarm 0 is enabled and the RTC0 timer value matches the Alarm 0 threshold value. The RTC0 can be configured as a reset source by writing a 1 to the RTC0REN bit. The RTC0 reset remains functional even when the device is in a low power mode. The state of the RESET pin is unaffected by this reset.

6.1.8. Software Reset

Firmware may force a reset by writing a 1 to the SWREN bit. The SWRF bit will read 1 following a firmware forced reset. The state of the RESET pin is unaffected by this reset.

6.1.9. Core Reset

The Core Reset is a firmware reset generated in the NVIC by setting the SYSRESETREQ bit.

6.1.10. PMU or Wake Reset

The PMU can issue a system reset whenever a pin wake event occurs. The WAKERF flag indicates when a PMU Wake reset occurs.

All RSTSRC0 registers may be locked against writes by setting the CLKRSTL bit in the LOCK0_PERIPHLOCK0 register to 1.

6.2. RSTSRC0 Registers

This section contains the detailed register descriptions for RSTSRC0 registers.

Register 6.1. RSTSRC0_RESETEN: System Reset Source Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RTC0MREN	ACC0MREN	LCD0MREN	UART0MREN	CPMREN	Reserved										
Type	RW	RW	RW	RW	RW	R										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					RTC0REN	CPFREN	CMP1REN	CMP0REN	SWREN	WDTREN	MCDREN	Reserved	VMONREN	Reserved	
Type	R				RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	1	0	0	0	0	0	1	0	1	1	1	1

Register ALL Access Address

RSTSRC0_RESETEN = 0x4002_C000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 6.1. RSTSRC0_RESETEN Register Bit Descriptions

Bit	Name	Function
31	RTC0MREN	**RTC0 Module Reset Enable.** Setting this bit to 1 causes all reset events to reset the RTC0 module if the corresponding bit in RESETEN[11:2] is set to 1. When cleared to 0, only a power-on-reset will reset this block.
30	ACC0MREN	**ACCTR0 Module Reset Enable.** Setting this bit to 1 causes all reset events to reset the Advanced Capture Counter (ACCTR0) module if the corresponding bit in RESETEN[11:2] is set to 1. When cleared to 0, only a power-on-reset will reset this block.
29	LCD0MREN	**LCD0 Module Reset Enable.** Setting this bit to 1 causes all reset events to reset the LCD0 module if the corresponding bit in RESETEN[11:2] is set to 1. When cleared to 0, only a power-on-reset will reset this block.
28	UART0MREN	**UART0 Module Reset Enable.** Setting this bit to 1 causes all reset events to reset the UART0 module if the corresponding bit in RESETEN[11:2] is set to 1. When cleared to 0, only a power-on-reset will reset this block.

Table 6.1. RSTSRC0_RESETEN Register Bit Descriptions

Bit	Name	Function
27	CPMREN	**Low Power Mode Charge Pump Module Reset Enable.** Setting this bit to 1 causes all reset events to reset the low power mode charge pump module if the corresponding bit in RESETEN[11:2] is set to 1. When cleared to 0, only a power-on-reset will reset this block.
26:11	Reserved	Must write reset value.
10	RTC0REN	**RTC0 Reset Enable.** 0: Disable the RTC0 event as a reset source. 1: Enable the RTC0 event as a reset source.
9	CPFREN	**Low Power Mode Charge Pump Supply Fail Reset Enable.** 0: Disable the low power mode charge pump supply fail event as a reset source. 1: Enable the low power mode charge pump supply fail event as a reset source.
8	CMP1REN	**Comparator 1 Reset Enable.** 0: Disable the Comparator 1 event as a reset source. 1: Enable the Comparator 1 event as a reset source.
7	CMP0REN	**Comparator 0 Reset Enable.** 0: Disable the Comparator 0 event as a reset source. 1: Enable the Comparator 0 event as a reset source.
6	SWREN	**Software Reset.** Writing a 1 to this bit generates a Software Reset.
5	WDTREN	**Watchdog Timer Reset Enable.** 0: Disable the Watchdog Timer event as a reset source. 1: Enable the Watchdog Timer event as a reset source.
4	MCDREN	**Missing Clock Detector Reset Enable.** 0: Disable the Missing Clock Detector event as a reset source. 1: Enable the Missing Clock Detector event as a reset source.
3	Reserved	Must write reset value.
2	VMONREN	**Voltage Supply Monitor VBAT Reset Enable.** 0: Disable the Voltage Supply Monitor VBAT event as a reset source. 1: Enable the Voltage Supply Monitor VBAT event as a reset source.
1:0	Reserved	Must write reset value.

Register 6.2. RSTSRC0_RESETFLAG: System Reset Flags

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				WAKERF	RTC0RF	CPFRF	CMP1RF	CMP0RF	SWRF	WDTRF	MCDRF	CORERF	VMONRF	PORRF	PINRF
Type	R				R	R	R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

RSTSRC0_RESETFLAG = 0x4002_C010

Table 6.2. RSTSRC0_RESETFLAG Register Bit Descriptions

Bit	Name	Function
31:12	Reserved	Must write reset value.
11	WAKERF	**PMU Wakeup Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A PMU Wakeup event did not cause the last system reset. 1: A PMU Wakeup event caused the last system reset.
10	RTC0RF	**RTC0 Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: An RTC0 event did not cause the last system reset. 1: An RTC0 event caused the last system reset.
9	CPFRF	**Low Power Mode Charge Pump Supply Fail Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A low power mode charge pump supply fail event did not cause the last system reset. 1: A low power mode charge pump supply fail event caused the last system reset.
8	CMP1RF	**Comparator 1 Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Comparator 1 event did not cause the last system reset. 1: A Comparator 1 event caused the last system reset.
7	CMP0RF	**Comparator 0 Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Comparator 0 event did not cause the last system reset. 1: A Comparator 0 event caused the last system reset.

Table 6.2. RSTSRC0_RESETFLAG Register Bit Descriptions

Bit	Name	Function
6	SWRF	**Software Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Software Reset event did not cause the last system reset. 1: A Software Reset event caused the last system reset.
5	WDTRF	**Watchdog Timer Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Watchdog Timer event did not cause the last system reset. 1: A Watchdog Timer event caused the last system reset.
4	MCDRF	**Missing Clock Detector Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Missing Clock Detector event did not cause the last system reset. 1: A Missing Clock Detector event caused the last system reset.
3	CORERF	**Core Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A Core Reset event did not cause the last system reset. 1: A Core Reset event caused the last system reset.
2	VMONRF	**Voltage Supply Monitor VBAT Reset Flag.** After checking PORRF, firmware should then check VMONRF for the last reset source, as all other flags are indeterminate if VMONRF is set. This flag is an indeterminate value when PORRF is set. 0: A Voltage Supply Monitor VBAT Reset event did not cause the last system reset. 1: A Voltage Supply Monitor VBAT Reset event caused the last system reset.
1	PORRF	**Power-On Reset Flag.** This flag should be checked first by firmware as all other flags are indeterminate if PORRF is set. This flag is an indeterminate value when VMONRF is set. 0: A Power-On Reset event did not cause the last system reset. 1: A Power-On Reset event caused the last system reset.
0	PINRF	**Pin Reset Flag.** This flag is an indeterminate value when VMONRF or PORRF is set. 0: A $\overline{\text{RESET}}$ pin event did not cause the last system reset. 1: A $\overline{\text{RESET}}$ pin event caused the last system reset.

6.3. RSTSRC0 Register Memory Map

Table 6.3. RSTSRC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RSTSRC0_RESETEN	0x4002_C000	ALL | SET | CLR	RTC0MREN	ACC0MREN	LCD0MREN	UART0MREN	CPMREN	Reserved																RTC0REN	CPFREN	CMP1REN	CMP0REN	SWREN	WDTREN	MCDREN	Reserved	VMONREN	Reserved	
RSTSRC0_RESETFLAG	0x4002_C010	ALL	Reserved																				WAKERF	RTC0RF	CPFRF	CMP1RF	CMP0RF	SWRF	WDTRF	MCDRF	CORERF	VMONRF	PORRF	PINRF

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

7. Register Security (LOCK0)

This section describes the Register Security (LOCK0) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

7.1. Security Features

The Security module includes the following features:

- Centralized global key protection implementation with mask bits for each peripheral group.

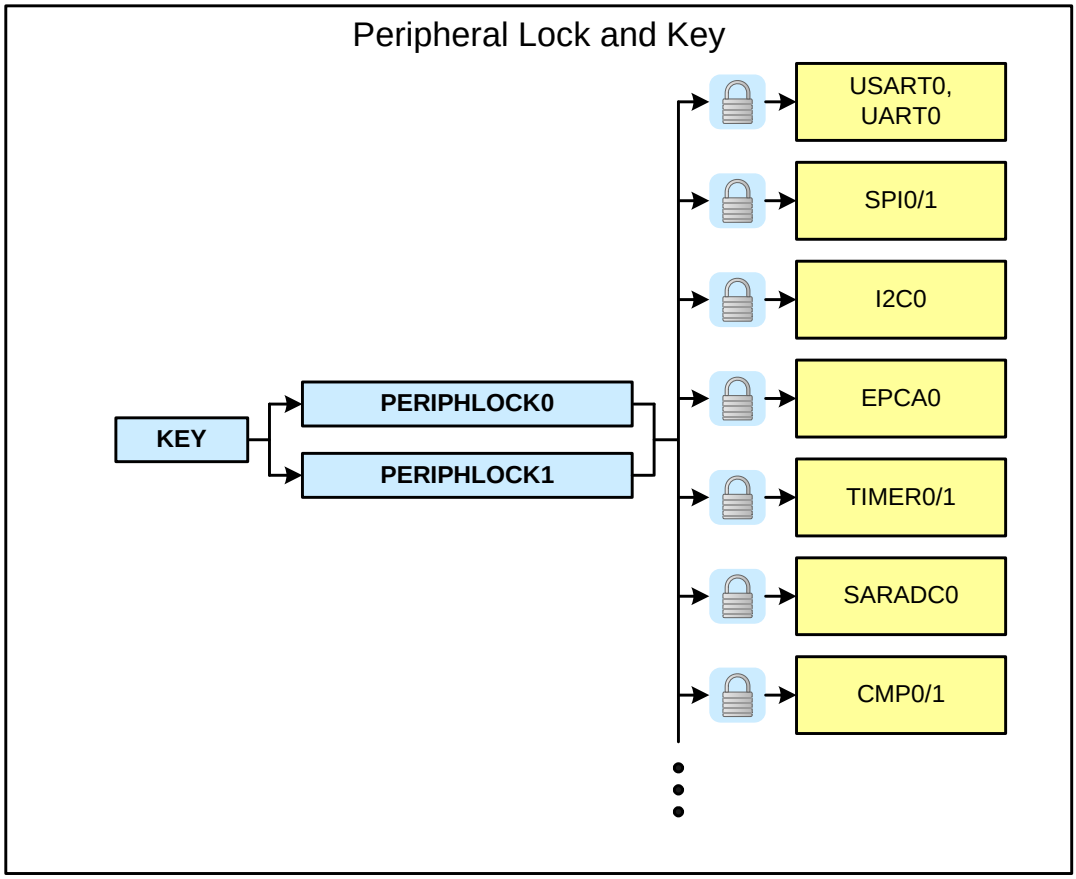


Figure 7.1. Security Block Diagram

The peripherals on the SiM3L1xx devices have a register lock and key mechanism that prevents any undesired accesses of the peripherals from firmware. Each bit in the PERIPHLOCKx registers controls a set of peripherals. A key sequence must be written to the KEY register to modify any of the bits in PERIPHLOCK0 or PERIPOHLOCK1. Any subsequent write to KEY will inhibit any accesses of the PERIPHLOCK registers until they are unlocked again through KEY. Reading the KEY register indicates the current status of the PERIPHLOCK registers lock state.

If a peripheral's registers are locked, all writes will be ignored. With the exception of the RTC0 and LPT0 modules, the registers can always be read, regardless of the peripheral's lock state. RTC0 and LPT0 require a running clock for registers to be read correctly.

7.2. LOCK0 Registers

This section contains the detailed register descriptions for LOCK0 registers.

Register 7.1. LOCK0_KEY: Security Key

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								KEY							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LOCK0_KEY = 0x4004_9000

Table 7.1. LOCK0_KEY Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:0	KEY	**Peripheral Lock Mask Key.** This field prevents accidental writes to the PERIPHLOCK0 and PERIPHLOCK1 registers. Writing 0xA5 followed by 0xF1 will unlock the PERIPHLOCK registers. Any value written to KEY while the PERIPHLOCK registers are unlocked will lock the register. Reading this register returns the current lock state (0 = Registers locked and no keys written, 1 = Registers locked and first key written, 2 = Registers unlocked)

Register 7.2. LOCK0_PERIPHLOCK0: Peripheral Lock Control 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	DCDCL	LCDL	DTML	PMUL	Reserved				ACCTRL	LPOSCL	PVTL	EXTOSCL	PLLL	LDOL	LPTL
Type	R	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DMAXBARL	DMACTRLL	IDACL	VMONL	CLKCTRL	RSTSRCL	RTCL	CRCL	AESL	CMPL	SARADCL	TIMERL	PCAL	I2CL	SPIL	USARTL
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LOCK0_PERIPHLOCK0 = 0x4004_9020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 7.2. LOCK0_PERIPHLOCK0 Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	DCDCL	**DC-DC Converter Module Lock.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the DCDC0 Module registers. 1: Lock the DCDC0 Module registers (bits can still be read).
29	LCDL	**LCD Module Lock.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the LCD0 Module registers. 1: Lock the LCD0 Module registers (bits can still be read).
28	DTML	**DTM Module Lock.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the DTM0, DTM1, and DTM2 Module registers. 1: Lock the DTM0, DTM1, and DTM2 Module registers (bits can still be read).
27	PMUL	**PMU Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the PMU Module registers. 1: Lock the PMU Module registers (bits can still be read).
26:23	Reserved	Must write reset value.

Table 7.2. LOCK0_PERIPHLOCK0 Register Bit Descriptions

Bit	Name	Function
22	ACCTRL	**Advanced Capture Counter Module Lock.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Advanced Capture Counter (ACCTR0) Module registers. 1: Lock the Advanced Capture Counter (ACCTR0) Module registers (bits can still be read).
21	LPOSCL	**Low Power Oscillator Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Low Power Oscillator (LPOSC0) Module registers. 1: Lock the Low Power Oscillator (LPOSC0) Module registers (bits can still be read).
20	PVTL	**PVT Oscillator Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the PVTOSC0 Module registers. 1: Lock the PVTOSC0 Module registers (bits can still be read).
19	EXTOSCL	**External Oscillator Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the External Oscillator (EXTOSC0) Module registers. 1: Lock the External Oscillator (EXTOSC0) Module registers (bits can still be read).
18	PLLL	**PLL Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the PLL0 Module registers. 1: Lock the PLL0 Module registers (bits can still be read).
17	LDOL	**Voltage Reference Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the LDO0 Module registers. 1: Lock the LDO0 Module registers (bits can still be read).
16	LPTL	**Low Power Timer Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Low Power Timer (LPTIMER0) Module registers. 1: Lock the Low Power Timer (LPTIMER0) Module registers (bits can still be read).
15	DMAXBARL	**DMA Crossbar Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the DMA Crossbar (DMAXBAR0) Module registers. 1: Lock the DMA Crossbar (DMAXBAR0) Module registers (bits can still be read).
14	DMACTRLL	**DMA Controller Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the DMA Controller (DMACTRL0) Module registers. 1: Lock the DMA Controller (DMACTRL0) Module registers (bits can still be read).
13	IDACL	**IDAC Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the IDAC0 Module registers. 1: Lock the IDAC0 Module registers (bits can still be read).

Table 7.2. LOCK0_PERIPHLOCK0 Register Bit Descriptions

Bit	Name	Function
12	VMONL	**Voltage Supply Monitor Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Voltage Supply Monitor (VMON0) Module registers. 1: Lock the Voltage Supply Monitor (VMON0) Module registers (bits can still be read).
11	CLKCTRL	**Clock Control Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Clock Control (CLKCTRL)Module registers. 1: Lock the Clock Control (CLKCTRL) Module registers (bits can still be read).
10	RSTSRCL	**Reset Sources Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Reset Sources (RSTSRC) Module registers. 1: Lock the Reset Sources (RSTSRC) Module registers (bits can still be read).
9	RTCL	**RTC Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the RTC0 Module registers. 1: Lock the RTC0 Module registers (bits can still be read).
8	CRCL	**CRC Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the CRC0 Module registers. 1: Lock the CRC0 Module registers (bits can still be read).
7	AESL	**AES Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the AES0 Module registers. 1: Lock the AES0 Module registers (bits can still be read).
6	CMPL	**Comparator Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the Comparator 0 and Comparator 1 Module registers. 1: Lock the Comparator 0 and Comparator 1 Module registers (bits can still be read).
5	SARADCL	**SARADC Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the SARADC0 Module registers. 1: Lock the SARADC0 Module registers (bits can still be read).
4	TIMERL	**Timer Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the TIMER0, TIMER1, and TIMER2 Module registers. 1: Lock the TIMER0, TIMER1, and TIMER2 Module registers (bits can still be read).

Table 7.2. LOCK0_PERIPHLOCK0 Register Bit Descriptions

Bit	Name	Function
3	PCAL	**PCA Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the EPCA0 Module registers. 1: Lock the EPCA0 Module registers (bits can still be read).
2	I2CL	**I2C Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the I2C0 Module registers. 1: Lock the I2C0 Module registers (bits can still be read).
1	SPIL	**SPI Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the SPI0 and SPI1 Module registers. 1: Lock the SPI0 and SPI1 Module registers (bits can still be read).
0	USARTL	**USART/UART Module Lock Enable.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the USART0 and UART0 Module registers. 1: Lock the USART0 and UART0 Module registers (bits can still be read).

Register 7.3. LOCK0_PERIPHLOCK1: Peripheral Lock Control 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															ENCDECL
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LOCK0_PERIPHLOCK1 = 0x4004_9040

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 7.3. LOCK0_PERIPHLOCK1 Register Bit Descriptions

Bit	Name	Function
31:1	Reserved	Must write reset value.
0	ENCDECL	**Encoder Decoder Module Lock.** This bit cannot be written until the PERIPHLOCK0 register is unlocked using KEY. 0: Unlock the ENCDEC0 Module registers. 1: Lock the ENCDEC0 Module registers (bits can still be read).

7.3. LOCK0 Register Memory Map

Table 7.4. LOCK0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LOCK0_KEY	0x4004_9000	ALL	Reserved																								KEY							
LOCK0_PERIPHLOCK0	0x4004_9020	ALL | SET | CLR	Reserved	DCDCL	LCDL	DTML	PMUL	Reserved				ACCTRL	LPOSCL	PVTL	EXTOSCL	PLLL	LDOL	LPTL	DMAXBARL	DMACTRLL	IDACL	VMONL	CLKCTRL	RSTSRCL	RTCL	CRCL	AESL	CMPL	SARADCL	TIMERL	PCAL	I2CL	SPIL	USARTL
LOCK0_PERIPHLOCK1	0x4004_9040	ALL | SET | CLR	Reserved																														ENCDECL

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

8. Port I/O Configuration

This section describes the Port I/O Crossbars and general Port Bank configuration and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

8.1. Port Bank Description

The features of the SiM3L1xx port banks are shown in Table 8.1.

Table 8.1. Port Bank Features

Port Bank	Type	Crossbar	LCD0	Pulse Generator	Supply Domain
PB0	Standard (PBSTD)	✓		✓	VIO
PB1	Standard (PBSTD)	✓	✓		VIO
PB2	Standard (PBSTD)	✓			VIORF
PB3	Standard (PBSTD)	PB3.0-PB3.7	✓		VIO
PB4	General Purpose (PBGP)		✓		VIO

8.2. Crossbar

8.2.1. Crossbar Features

A port I/O crossbar is included on the SiM3L1xx, with the following features:

- Flexible assignment of many digital peripherals to port pins.
- Pin skip capabilities to reserve specific I/O for other purposes (analog signals, GPIO, layout considerations)

The port I/O crossbar on the SiM3L1xx is used to route many of the digital peripherals to the device I/O pins. It allows the user to select the specific mix of peripherals that are needed for the application, and route them out of the device, leaving the unused pins available for general-purpose I/O. The crossbar maps peripherals to port pins PB0.0 through PB3.7 on all devices. PB3.8-PB3.15 and PB4 pins are not available to the crossbar.

8.2.2. Crossbar Configuration

The peripherals which are routed through the crossbar have a specific priority order when assigned to pins and a specific range of pins where they can be routed. The crossbar assigns peripherals in priority order, starting with the highest-priority peripherals and the lowest-order port pins available to the peripherals. When a peripheral is enabled, all of the pins associated with that peripheral are routed in sequence. Some peripherals are split into multiple functional groups, to route only the necessary pins. Additionally, pin skip registers can be used to prevent the crossbar from assigning peripherals to those pins.

When configuring the crossbar, all settings should be made to the crossbar and Port Bank registers before enabling the crossbar. This ensures that peripherals will not shift around while each one is being enabled and Port I/O pins will remain stable. The settings in PBOUTMD, PBMDSEL, or PBSKIPEN will not take effect until the crossbars are enabled.

If any pins are used for special functions not associated with the crossbars, these pins should be skipped using the corresponding Port Bank PBSKIPEN register. This applies to External Interrupts, SARADC0 inputs, Comparator inputs, IDAC0 outputs, LCD0 signals and other analog and non-crossbar signals.

Register XBAR0 is used to enable peripherals on the crossbar. Some peripherals use multiple signals that are all enabled when the corresponding bit is set to 1. For example, enabling I2C0 on the crossbar enables both data and clock signals (SDA and SCL). Several peripherals have individual groups of pins that can be enabled or disabled as well. In the crossbar priority table (Table 8.2), such pins are listed with the same numerical priority and a letter (A, B, or C) indicating the priority of the functional group. In all cases, the primary functional group (A) must be enabled on the crossbar for the other groups (B or C) to be routed out. For example, the USART0 Data group (priority 1A) must be enabled in order to use either the USART0 Flow Control group (priority 1B) or the Clock Output group (priority 1C).

8.2.2.1. Crossbar 0

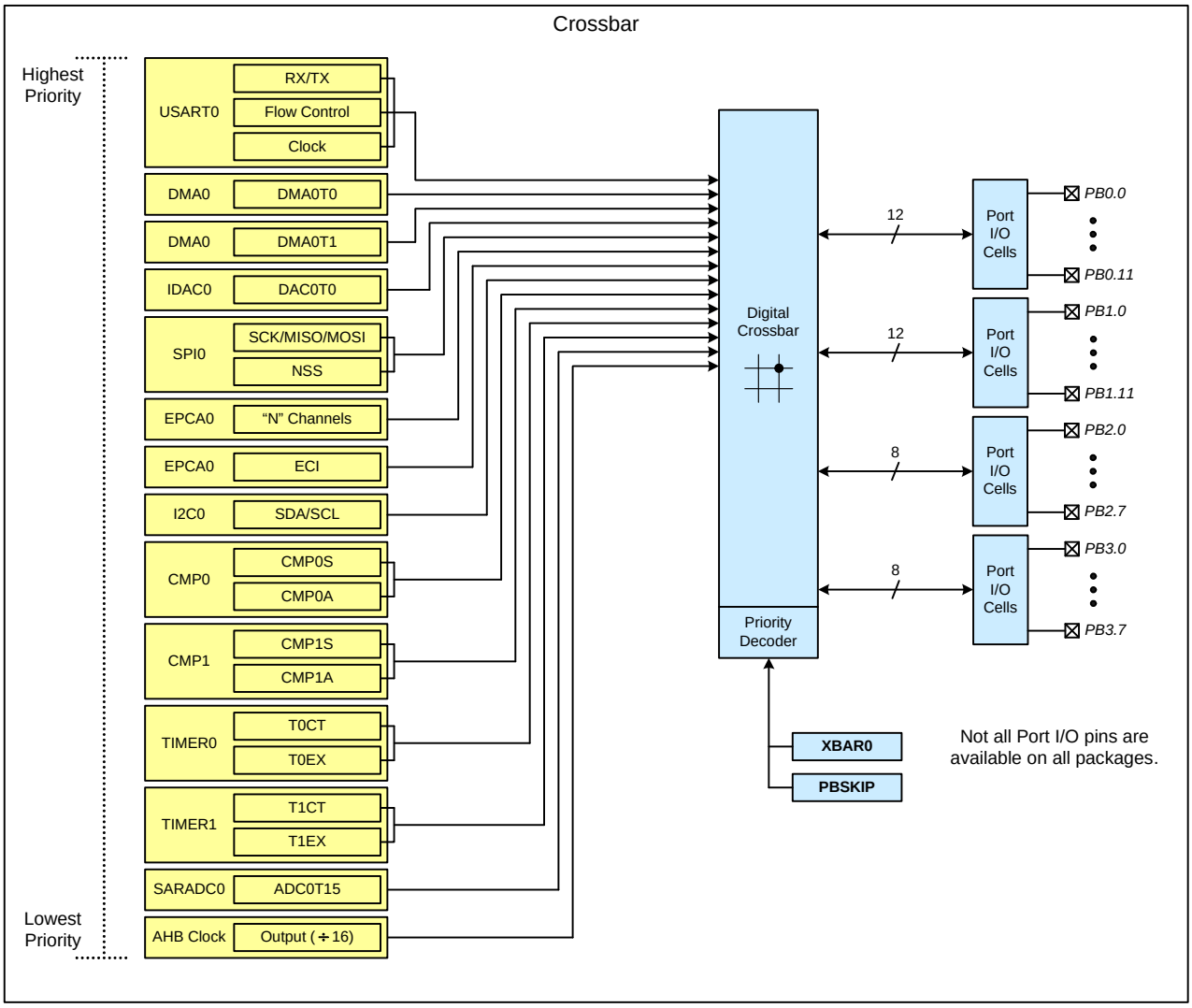


Figure 8.1. Crossbar 0 Block Diagram

Table 8.2. Crossbar 0 Peripherals and Priority

Peripheral Name	Functional Group	Priority	Enable Bits	Signal Names (in order)
USART0	Data	1-A	USART0EN	USART0_TX USART0_RX
	Flow Control	1-B	USART0FCEN	USART0_RTS USART0_CTS
	Clock Out	1-C	USART0CEN	USART0_UCLK
DMA0	Trigger Input 0	2	DMA0T0EN	DMA0T0
	Trigger Input 1	3	DMA0T1EN	DMA0T1
IDAC0	Trigger Input	4	IDAC0TEN	DAC0T0
SPI0	Clock/Data	5-A	SPI0EN	SPI0_SCK SPI0_MISO SPI0_MOSI
	Slave Select	5-B	SPI0NSSEN	SPI0_NSS
EPCA0	Input / Output	6	EPCA0EN[2:0]	EPCA0_STD_CEX0 EPCA0_STD_CEX1 EPCA0_STD_CEX2 EPCA0_STD_CEX3 EPCA0_STD_CEX4 EPCA0_STD_CEX5
EECI0	Clock Input	7	EECI0EN	EPCA0_ECI
I2C0	Data	8	I2C0EN	I2C0_SDA I2C0_SCL
Comparator 0	Synchronous Output	9	CMP0SEN	CMP0_S
	Asynchronous Output	10	CMP0AEN	CMP0_A
Comparator 1	Synchronous Output	11	CMP1SEN	CMP1_S
	Asynchronous Output	12	CMP1AEN	CMP1_A
Timer 0	Count	13	TMR0CTEN	TIMER0_CT
	Input / Output	14	TMR0EXEN	TIMER0_EX
Timer 1	Count	15	TMR1CTEN	TIMER1_CT
	Input / Output	16	TMR1EXEN	TIMER1_EX
SARADC0	Trigger Input	17	SARADC0TEN	ADC0T15
AHB Clock / 16	Output	18	AHBEN	AHB_OUT

8.2.2.2. Crossbar Configuration Example

Configuring Crossbar 0 to enable the USART0 data pins, and three EPCA0 channels:

		P0												P1												P2								P3							
Peripheral	Signal Name	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	8	9	10	11	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
USART0	USART0_TX																																								
	USART0_RX																																								
	USART0_RTS																																								
	USART0_CTS																																								
	USART0_UCLK																																								
DMA0	DMA0T0																																								
	DMA0T1																																								
IDAC0	DAC0T0																																								
SPI0	SPI0_SCK																																								
	SPI0_MISO																																								
	SPI0_MOSI																																								
	SPI0_NSS																																								
EPCA0	EPCA0_CEX0																																								
	EPCA0_CEX1																																								
	EPCA0_CEX2																																								
	EPCA0_CEX3																																								
	EPCA0_CEX4																																								
	EPCA0_CEX5																																								
EPCA0 ECI	EPCA0_ECI																																								
I2C0	I2C0_SDA																																								
	I2C0_SCL																																								
CMP0	CMP0S																																								
	CMP0A																																								
CMP1	CMP1S																																								
	CMP1A																																								
TIMER0	TIMER0_CT																																								
	TIMER0_EX																																								
TIMER1	TIMER1_CT																																								
	TIMER1_EX																																								
SARADC0	ADC0T15																																								
AHB Clock / 16	AHB_OUT																																								
	PBSKIPEN	1	1	1	1	1	1	1	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.2. Crossbar 0 Example Configuration

In this particular example, PB0.0-PB0.7, PB0.10, and PB0.11 are skipped. As a result, the USART0 pins are placed on PB0.8-PB0.9, since this is the highest priority peripheral. The EPCA0 signals are next, and they're placed on PB1.0-PB1.2, since PB0.10 and PB0.11 are skipped.

8.3. Port Bank Standard (PBSTD) and Port Bank General Purpose (PBGP) Features

The Port Bank modules include the following features:

- Push-pull or open-drain output modes and analog or digital input modes.
- Option for high or low output drive strength.
- Port Match allows any pin or combination of pins to generate an interrupt.
- Internal pull-up resistors are enabled or disabled on a port-by-port basis.
- Pulse generator logic which can produce fast pulses on one or more output pins (PB0 only).
- Peripheral crossbar assignment on PBSTD blocks (PB0, PB1, PB2 and PB3).

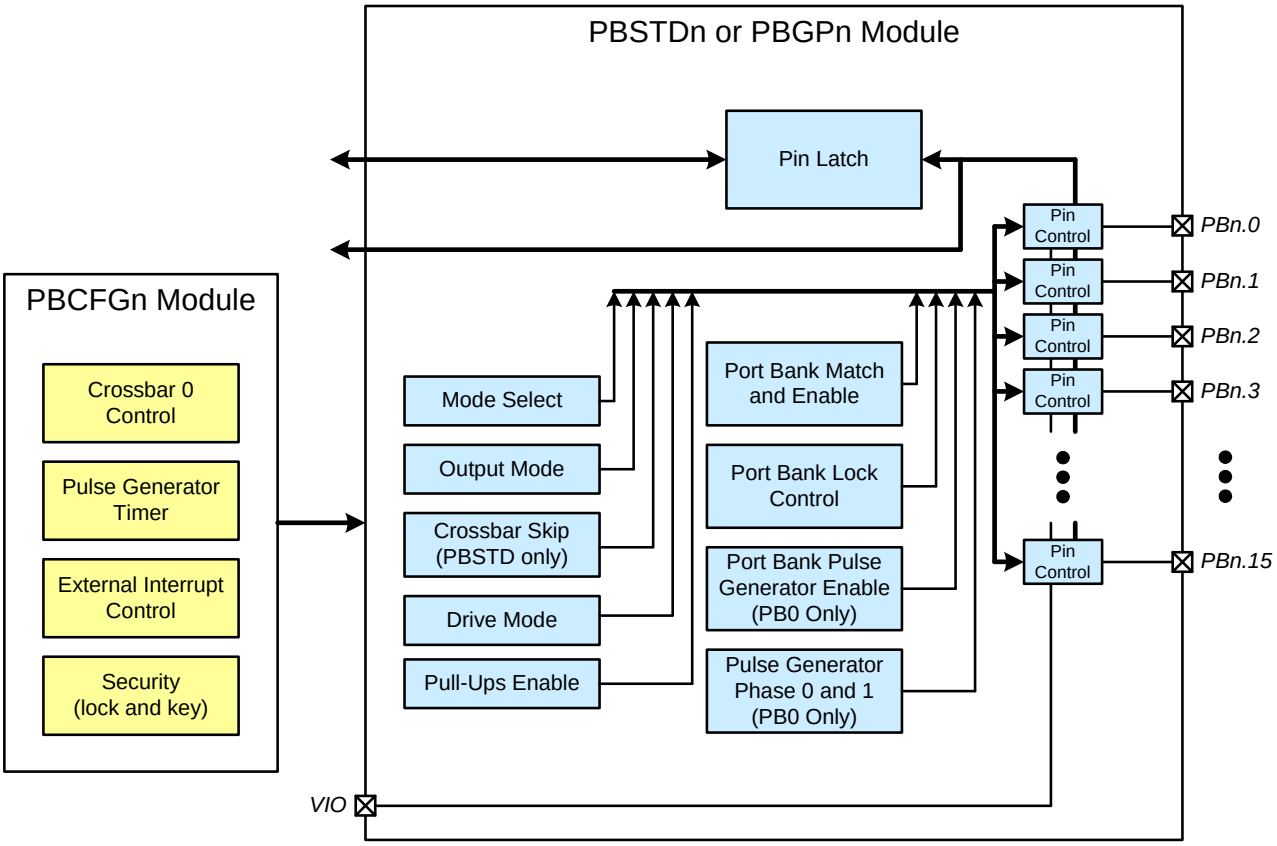


Figure 8.3. PBSTD and PBGP Block Diagram

8.4. Standard Modes of Operation

Each Port Bank pin can be configured by firmware for analog I/O or digital I/O using the PBMDSEL register. On reset, all Port Bank cells default to a digital high impedance state with weak pull-ups enabled.

8.4.1. Port Bank Pins Configured for Analog I/O

Any pins used for an analog function should be configured for analog I/O (PBMDSEL.x = 0). When a pin is configured for analog I/O, its weak pullup and digital receiver are disabled. In most cases, firmware should also disable the digital output drivers. Firmware will always read back a value of 0 from the PBPIN register for port pins configured for analog I/O regardless of the actual voltage on the pin.

Configuring pins as analog I/O saves power and isolates the port pin from digital interference. Port pins configured as digital inputs may still be used by analog peripherals; however, this practice is not recommended and may result in measurement errors.

8.4.2. Port Pins Configured For Digital I/O

Any pins used by digital peripherals (USART, SPI, I2C, etc.), external digital event capture functions, or as GPIO should be configured as digital I/O (PBMDSEL.x = 1). For digital I/O pins, one of two output modes (push-pull or open-drain) must be selected using the PBOUTMD register.

Push-pull outputs (PBOUTMD.x = 1) drive the port pad to the high or low supply rail based on the output logic value of the port pin. Open-drain outputs have the high side driver disabled; therefore, they only drive the port pad to VSS when the output logic value is 0 and become high impedance (both high and low drivers turned off) when the output logic value is 1.

Note: The port output drivers for pins on the crossbar are disabled when the crossbar is disabled. The crossbar must be enabled before using any pin as a general-purpose output.

When a digital I/O cell is placed in the high impedance state, a weak pull-up transistor pulls the port pad to the VIO (or VIORF) supply voltage to ensure the digital input is at a defined logic state. Weak pull-ups are disabled when the I/O cell is driven to VSS to minimize power consumption and may be disabled on a port-by-port basis by clearing PBPUEN to 0. The user should ensure that digital I/O are always internally or externally pulled or driven to a valid logic state to prevent extra supply current caused by intermediate values. From the PB register, port pins configured for digital I/O always read back the logic state last written to the latch, regardless of the output logic value of the port pin. The sensed output logic value of the pins (high or low) can be read using the PBPIN register.

8.4.3. Increasing Port I/O Drive Strength

Port Bank output drivers support a high and low drive strength; the default is low drive strength. The drive strength of a pin is configurable using the PBDRV register. See the electrical specifications chapter for the difference in output drive strength between the two modes.

8.4.4. Interfacing Port I/O to 5 V Logic

Port Bank pins configured for digital, open-drain operation are capable of interfacing to digital logic operating above the supply pin. To provide logic high “output” to external systems above the rail, an external pullup resistor is required.

Important Note: In a multi-voltage interface, the external pull-up resistor should be sized to allow a current of at least 150 μA to flow into the port pin when the pin voltage is between V_{IO} + 0.4 V and V_{IO} + 1.0 V. When the pin voltage increases beyond this range, the current flowing into the port pin is minimal.

8.5. Assigning Standard Port Bank Pins to Analog and Digital Functions

Port Bank pins can be assigned to various analog, digital, and external interrupt functions. The port pins assigned to analog functions should be configured for analog I/O and port pins assigned to digital or external interrupt functions should be configured for digital I/O.

8.5.1. Assigning Port Bank Pins to Analog Functions

Port pins selected for analog functions should have their digital drivers disabled (PBOUTMD.x = 0 and PB.x = 1) and their corresponding bit in PBSKIPEN set to 1. This reserves the pin for use by the analog function and does not allow it to be claimed by the crossbar.

8.5.2. Assigning Port Bank Pins to Digital Functions

Any Port Bank pins not assigned to analog functions may be assigned to digital functions or used as GPIO. Most digital functions rely on the crossbar for pin assignment; however, some digital functions bypass the crossbar. Port pins used by these digital functions and any port pins selected for use as GPIO or I/O pins not available in the package being used should have their corresponding bit in PBSKIPEN set to 1.

8.5.3. Assigning Port Bank Pins to External Digital Event Capture Functions

External digital event capture functions can be used to trigger an interrupt or wake the device from a low power mode when a transition occurs on a digital I/O pin. The digital event capture functions do not require dedicated pins and will function on both GPIO pins (PBSKIPEN.x = 1) and pins in use by the Crossbars (PBSKIPEN.x = 0). External digital event capture functions cannot be used on pins configured for analog I/O.

8.6. Port Match

Port Match functionality allows system events to be triggered by a logic value change on any port pin. A port match event occurs if the logic levels of any of the selected input pins match the firmware controlled value in the PM register. This allows firmware to be notified if a certain change occurs on input pins regardless of the crossbar settings.

The PMEN registers can be used to individually select which Port Bank pins should be compared against the PM registers.

A port match event may be used to generate an interrupt. If multiple pins are used to generate a port match event, the individual pins can be checked inside the Port Match ISR to determine the cause of the current interrupt. In this case, the event that causes the interrupt must be present long enough to enter the ISR and check the current state of the pins.

8.7. Pulse Generator

PB0 on SiM3L1xx devices has an additional Pulse Generator feature. This Pulse Generator provides the capability to toggle pins of the PB0 port from a single 32-bit word write with a preset 5-bit delay between the setting and clearing (or clearing and setting) of the pins. This 5-bit delay is implemented as a timer in the PGTIMER field where the actual count time is PGTIMER + 1. This timer is clocked from the APB clock, and the PGDONEF bit asserts when the timer expires.

Writing to the PBPGPHASE register will update the PB0 value with the phase 0 (PBPGPH0) value, start the counter, and clear the PGDONEF bit. When the timer expires, the PB0 register updates with the phase 1 (PBPGPH1) value, and PGDONEF is set. The PBPGEN register selects which pins are controlled by the pulse generator.

For example:

1. Set PB0's mask register PBPGEN to 0x000000FF.
2. Write the 5-bit pulse generator timer value, PGTIMER, to 0x0F for a 16 cycle delay.
3. Write the PBPGPHASE register with 0x000500FF to set PBPGPH0 and PBPGPH1 at the same time.
4. Poll for PGDONEF.

In this example, the Pulse Generator writes PB0[7:0] to 0xFF, waits 16 cycles, and then updates PB0[7:0] to 0x05. PB0[14:8] are unaffected by this operation since the PBPGEN register only enables bits [7:0] for a pulse generation update.

While PGDONEF is zero, the PB0 and PBPGPHASE bits that have been enabled for pulse generation with the PBPGEN register cannot be updated with a register write. In the previous example, any writes to PB0 during the 16 cycles will take according to the restrictions imposed by the PBPGEN register. For example, if PB0[14:8] was written to 0x5A, this write will take effect since PBPGEN[14:8] was set to 0x00. Conversely, if PB0[7:0] was written during this period, it will not take effect since PBPGEN[7:0] was set to 0xFF.

8.8. Port Bank Security

The port bank control registers have a locking mechanism, controlled by the LOCK bit. Control and Configuration registers are locked if the LOCK bit is set and the lock is in a locked state. This lock ensures the fields that control the peripheral output mapping cannot change inadvertently. Setting LOCK will lock CONTROL1, XBAR0, and all PBSKIPEN registers.

If the control registers are locked (LOCK set to 1), the sequence of 0xA5 followed by 0xF1 must be written to the PBKEY register. Once unlocked, writing to a lockable register or writing an any value to PBKEY will relock the registers. All of the registers can be read at any time, regardless of the lock state. Additionally, writes to non-lockable registers (like CONTROL0) do not affect the state of the lock. The PBKEY register can be read at any time to return the status of the lock.

8.9. Debugging Interfaces

After a reset, SiM3L1xx devices are configured with JTAG enabled to allow external JTAG modules to connect. Firmware must disable this if not needed, freeing the JTAG pins for use with other functions. If the core is configured for Serial Wire (SW) mode and not JTAG, then the Serial Wire Viewer is enabled to come out of the TDO pin and the TDI pin is available for other Crossbar or GPIO functions. The Serial Wire Viewer (SWV) provides a single pin to send out TPIU messages.

Enabling the JTAG or ETM interfaces overrides all other Crossbar and GPIO functionality, so these pins should be skipped in the PBSKIPEN registers, and all writes to these bits in PB will be ignored. ETM must also be enabled in the core. Note that when the Serial Wire Viewer (SWV) is enabled, the associated port I/O pin should be configured in open-drain digital mode.

Table 8.3. Debug Interface Pin Information

Debug Interface	Debug Signal Name	Conditions	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
JTAG	TCK	JTAGEN = 1	SWCLK/TCK		
	TDO		PB0.11		
	TDI		PB1.6		
	TMS		SWDIO/TMS		
ETM	ETM0	ETMEN = 1	PB4.14	PB4.11	
	ETM1		PB4.13	PB4.10	
	ETM2		PB4.12	PB4.9	
	ETM3		PB4.11	PB4.8	
	TRACECLK		PB4.15	PB4.12	
Serial Wire Debug	SWCLK	Clock sequence on SWCLK	SWCLK/TCK	SWCLK	SWCLK
	SWDIO		SWDIO/TMS	SWDIO	SWDIO
Serial Wire Viewer	SWV	SWVEN = 1	PB0.11	PB0.9	PB0.6

8.10. External Interrupts

8.10.1. External Interrupt Features

The External Interrupts (INT0/INT1) on the SiM3L1xx devices have the following features:

- Level (high or low) or edge (rising, falling, or both) detection.
- Can select one of up to 16 inputs to monitor.
- Separate from the crossbar, and can be used in conjunction with other peripherals on the same pins.

The INT0 and INT1 external interrupt sources are configurable as active high or low, edge or level sensitive. The INT0POL (INT0 Polarity) and INT1POL (INT1 Polarity) bits in the CONTROL0 register select active high or active low; the INT0MD and INT1MD fields in the same register select level or edge sensitive modes. When both edge mode is selected, the polarity selection is ignored. The table below lists the possible configurations.

Table 8.4. External Interrupt Configuration

INTnMD	INTnPOL	INTn Configuration
00	0	Active low, level sensitive (low level)
00	1	Active high, level sensitive (high level)
01	0	Active low, edge sensitive (falling)
01	1	Active high, edge sensitive (rising)
10	X	Both edges (rising and falling)

INT0 and INT1 are assigned to Port Bank pins as defined in the CONTROL0 register. These pin assignments are independent of any crossbar assignments. The External Interrupts will monitor their assigned Port Bank pins without disturbing the peripheral that was assigned to the pin via the crossbars. To assign a Port Bank pin only to INT0 or INT1, configure the crossbar to skip the selected pins by setting the associated bit in PBSKIPEN register. The pins available for use as external interrupt sources vary by package, and are defined in Table 8.5.

Table 8.5. External Interrupt Triggers

Trigger	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
INT0.0	PB0.0	PB0.0	PB0.0
INT0.1	PB0.1	PB0.1	PB0.1
INT0.2	PB0.2	PB0.2	PB0.2
INT0.3	PB0.3	PB0.3	PB0.3
INT0.4	PB0.4	PB0.4	PB0.4
INT0.5	PB0.5	PB0.5	PB0.5
INT0.6	PB0.6	PB0.6	PB0.6
INT0.7	PB0.7	PB0.7	PB0.7
INT0.8	PB0.8	PB0.8	PB0.8

Table 8.5. External Interrupt Triggers

Trigger	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
INT0.9	PB0.9	PB0.9	PB0.9
INT0.10	PB0.10	Reserved	Reserved
INT0.11	PB0.11	Reserved	Reserved
INT0.12	PB1.0	PB1.0	Reserved
INT0.13	PB1.1	PB1.1	Reserved
INT0.14	PB1.2	PB1.2	Reserved
INT0.15	PB1.3	PB1.3	Reserved
INT1.0	PB2.0	PB2.0	PB2.0
INT1.1	PB2.1	Reserved	PB2.1
INT1.2	Reserved	Reserved	PB2.2
INT1.3	Reserved	Reserved	PB2.3
INT1.4	PB2.4	PB2.4	PB2.4
INT1.5	PB2.5	PB2.5	PB2.5
INT1.6	PB2.6	PB2.6	PB2.6
INT1.7	PB2.7	PB2.7	PB2.7
INT1.8	PB3.0	PB3.0	PB3.0
INT1.9	PB3.1	PB3.1	PB3.1
INT1.10	PB3.2	PB3.2	PB3.2
INT1.11	PB3.3	PB3.3	PB3.3
INT1.12	PB3.4	PB3.4	PB3.4
INT1.13	PB3.5	PB3.5	PB3.5
INT1.14	PB3.6	PB3.6	PB3.6
INT1.15	PB3.7	PB3.7	PB3.7

If an INT0 or INT1 external interrupt is configured as edge-sensitive, the corresponding interrupt-pending flag is set once per edge. When configured as level sensitive, the interrupt-pending flag remains logic 1 while the input is active as defined by the corresponding polarity bit (INT0POL or INT1POL); the flag remains cleared while the input is inactive. The external interrupt source must hold the input active until the interrupt request is recognized. It must then deactivate the interrupt request before execution of the ISR completes or another interrupt request will be generated.

8.11. PBCFG0 Registers

This section contains the detailed register descriptions for PBCFG0 registers.

Register 8.1. PBCFG0_CONTROL0: Global Port Control 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PGDONEF	Reserved		PGTIMER					Reserved							
Type	R	R		RW					R							
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	INT1EN	INT1MD		INT1POL	INT1SEL				INT0EN	INT0MD		INT0POL	INT0SEL			
Type	RW	RW		RW	RW				RW	RW		RW	RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBCFG0_CONTROL0 = 0x4002_A000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.6. PBCFG0_CONTROL0 Register Bit Descriptions

Bit	Name	Function
31	PGDONEF	**Pulse Generator Timer Done Flag.** This bit is cleared by hardware when firmware writes to the PBPGPHASE register. This bit is set when the Pulse Generator timer expires. While PGDONEF is 0, the PB and PBPGPHASE bits that have been enabled for pulse generation with the PBPGMSK cannot be updated with a register write.
30:29	Reserved	Must write reset value.
28:24	PGTIMER	**Pulse Generator Timer.** Count down timer value for supported toggle ports.
23:16	Reserved	Must write reset value.
15	INT1EN	**External Interrupt 1 Enable.** 0: Disable external interrupt 1. 1: Enable external interrupt 1.

Table 8.6. PBCFG0_CONTROL0 Register Bit Descriptions

Bit	Name	Function
14:13	INT1MD	**External Interrupt 1 Mode.** 00: Interrupt on logic level at pin, as selected by the INT1POL field. 01: Interrupt on either rising or falling edge, as selected by the INT1POL field. 10: Interrupt on both rising and falling edges (ignores INT1POL). 11: Reserved.
12	INT1POL	**External Interrupt 1 Polarity.** 0: A low value or falling edge on the selected pin will cause interrupt. 1: A high value or rising edge on the selected pin will cause interrupt.
11:8	INT1SEL	**External Interrupt 1 Pin Selection.** Selects the external pin to use as external interrupt 1. (INT1SEL = 0 selects INT1.0).
7	INT0EN	**External Interrupt 0 Enable.** 0: Disable external interrupt 0. 1: Enable external interrupt 0.
6:5	INT0MD	**External Interrupt 0 Mode.** 00: Interrupt on logic level at pin, as selected by the INT0POL field. 01: Interrupt on either rising or falling edge, as selected by the INT0POL field. 10: Interrupt on both rising and falling edges (ignores INT0POL). 11: Reserved.
4	INT0POL	**External Interrupt 0 Polarity.** 0: A low value or falling edge on the selected pin will cause interrupt. 1: A high value or rising edge on the selected pin will cause interrupt.
3:0	INT0SEL	**External Interrupt 0 Pin Selection.** Selects the external pin to use as external interrupt 0 (INT0SEL = 0 selects INT0.0).

Register 8.2. PBCFG0_CONTROL1: Global Port Control 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LOCK	Reserved														
Type	RW	R														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			LPTOSEL	Reserved		PMATCHEN	SPI1SEL	Reserved					SWVEN	ETMEN	JTAGEN
Type	R			RW	R		RW	RW	RW					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register ALL Access Address

PBCFG0_CONTROL1 = 0x4002_A010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.7. PBCFG0_CONTROL1 Register Bit Descriptions

Bit	Name	Function
31	LOCK	**Port Bank Configuration Lock.** This bit controls the lock for the port bank configuration and control registers. 0: Port Bank Configuration and Control registers are unlocked. 1: The following registers are locked from write access: CONTROL1, XBAR0, and all PBSKIP registers.
30:13	Reserved	Must write reset value.
12	LPTOSEL	**Low Power Timer Output Pin Select.** 0: Route the Low Power Timer output to LPT0OUT0. 1: Route the Low Power Timer output to LPT0OUT1.
11:10	Reserved	Must write reset value.
9	PMATCHEN	**Port Match Interrupt Enable.** 0: Disable the port match logic. The PBnMAT registers are not read/write accessible on the APB bus. 1: Enable the port match logic to generate a port match interrupt. The PBnMAT registers are read/write accessible on the APB bus.
8	SPI1SEL	**SPI1 Fixed Port Selection.** 0: Disconnect SPI1 from the dedicated pins. 1: Connect SPI1 to the dedicated pins.
7:3	Reserved	Must write reset value.

Table 8.7. PBCFG0_CONTROL1 Register Bit Descriptions

Bit	Name	Function
2	SWVEN	**SWV Enable.** 0: SWV is not pinned out. 1: SWV is enabled and pinned out.
1	ETMEN	**ETM Enable.** 0: ETM not pinned out. 1: ETM is enabled and pinned out.
0	JTAGEN	**JTAG Enable.** 0: JTAG functionality is not pinned out. 1: JTAG functionality is pinned out.

Register 8.3. PBCFG0_XBAR0: Crossbar 0 Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	XBAR0EN	Reserved								AHBEN	SARADC0TEN	TMR1EXEN	TMR1CTEN	TMR0EXEN	TMR0CTEN	CMP1AEN
Type	RW	R								RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CMP1SEN	CMP0AEN	CMP0SEN	I2C0EN	EECI0EN	EPCA0EN			SPI0NSSEN	SPI0EN	IDAC0TEN	DMA0T1EN	DMA0T0EN	USART0CEN	USART0FCEN	USART0EN
Type	RW	RW	RW	RW	RW	RW			RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBCFG0_XBAR0 = 0x4002_A020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.8. PBCFG0_XBAR0 Register Bit Descriptions

Bit	Name	Function
31	XBAR0EN	**Crossbar 0 Enable.** 0: Disable Crossbar 0. 1: Enable Crossbar 0.
30:23	Reserved	Must write reset value.
22	AHBEN	**AHB Clock Output Enable.** 0: Disable the AHB Clock / 16 output on Crossbar 0. 1: Enable the AHB Clock / 16 output on Crossbar 0.
21	SARADC0TEN	**SARADC0 Trigger Enable.** 0: Disable SARADC0 conversion start trigger on Crossbar 0. 1: Enable SARADC0 conversion start trigger on Crossbar 0.
20	TMR1EXEN	**TIMER1 T1EX Enable.** 0: Disable TIMER1 EX on Crossbar 0. 1: Enable TIMER1 EX on Crossbar 0.

Table 8.8. PBCFG0_XBAR0 Register Bit Descriptions

Bit	Name	Function
19	TMR1CTEN	**TIMER1 T1CT Enable.** 0: Disable TIMER1 CT on Crossbar 0. 1: Enable TIMER1 CT on Crossbar 0.
18	TMR0EXEN	**TIMER0 T0EX Enable.** 0: Disable TIMER0 EX on Crossbar 0. 1: Enable TIMER0 EX on Crossbar 0.
17	TMR0CTEN	**TIMER0 T0CT Enable.** 0: Disable TIMER0 CT on Crossbar 0. 1: Enable TIMER0 CT on Crossbar 0.
16	CMP1AEN	**Comparator 1 Asynchronous Output (CMP1A) Enable.** 0: Disable Comparator 1 Asynchronous Output (CMP1A) on Crossbar 0. 1: Enable Comparator 1 Asynchronous Output (CMP1A) on Crossbar 0.
15	CMP1SEN	**Comparator 1 Synchronous Output (CMP1S) Enable.** 0: Disable Comparator 1 Synchronous Output (CMP1S) on Crossbar 0. 1: Enable Comparator 1 Synchronous Output (CMP1S) on Crossbar 0.
14	CMP0AEN	**Comparator 0 Asynchronous Output (CMP0A) Enable.** 0: Disable Comparator 0 Asynchronous Output (CMP0A) on Crossbar 0. 1: Enable Comparator 0 Asynchronous Output (CMP0A) on Crossbar 0.
13	CMP0SEN	**Comparator 0 Synchronous Output (CMP0S) Enable.** 0: Disable Comparator 0 Synchronous Output (CMP0S) on Crossbar 0. 1: Enable Comparator 0 Synchronous Output (CMP0S) on Crossbar 0.
12	I2C0EN	**I2C0 Enable.** 0: Disable I2C0 SDA and SCL on Crossbar 0. 1: Enable I2C0 SDA and SCL on Crossbar 0.
11	EECI0EN	**EPCA0 ECI Enable.** 0: Disable EPCA0 ECI on Crossbar 0. 1: Enable EPCA0 ECI on Crossbar 0.
10:8	EPCA0EN	**EPCA0 Channel Enable.** 000: Disable all EPCA0 channels on Crossbar 0. 001: Enable EPCA0 CEX0 on Crossbar 0. 010: Enable EPCA0 CEX0 and CEX1 on Crossbar 0. 011: Enable EPCA0 CEX0, CEX1, and CEX2 on Crossbar 0. 100: Enable EPCA0 CEX0, CEX1, CEX2, and CEX3 on Crossbar 0. 101: Enable EPCA0 CEX0, CEX1, CEX2, CEX3, and CEX4 on Crossbar 0. 110: Enable EPCA0 CEX0, CEX1, CEX2, CEX3, CEX4, and CEX5 on Crossbar 0. 111: Reserved.
7	SPI0NSSEN	**SPI0 NSS Pin Enable.** 0: Disable SPI0 NSS on Crossbar 0. 1: Enable SPI0 NSS on Crossbar 0.

Table 8.8. PBCFG0_XBAR0 Register Bit Descriptions

Bit	Name	Function
6	SPI0EN	**SPI0 Enable.** 0: Disable SPI0 SCK, MISO, and MOSI on Crossbar 0. 1: Enable SPI0 SCK, MISO, and MOSI on Crossbar 0.
5	IDAC0TEN	**IDAC0 Trigger Enable.** 0: Disable the IDAC0 trigger on Crossbar 0. 1: Enable the IDAC0 trigger on Crossbar 0.
4	DMA0T1EN	**DMA Trigger 1 Enabled.** 0: Disable the DMA trigger 1 on Crossbar 0. 1: Enable the DMA trigger 1 on Crossbar 0.
3	DMA0T0EN	**DMA Trigger 0 Enable.** 0: Disable the DMA trigger 0 on Crossbar 0. 1: Enable the DMA trigger 0 on Crossbar 0.
2	USART0CEN	**USART0 Clock Signal Enable.** 0: Disable USART0 clock on Crossbar 0. 1: Enable USART0 clock on Crossbar 0.
1	USART0FCEN	**USART0 Flow Control Enable.** 0: Disable USART0 flow control on Crossbar 0. 1: Enable USART0 flow control on Crossbar 0.
0	USART0EN	**USART0 Enable.** 0: Disable USART0 RX and TX on Crossbar 0. 1: Enable USART0 RX and TX on Crossbar 0.

Register 8.4. PBCFG0_PBKEY: Global Port Key

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								KEY							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBCFG0_PBKEY = 0x4002_A030

Table 8.9. PBCFG0_PBKEY Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:0	KEY	**Port Bank Key.** When the Port Banks are locked for access, firmware must write the value 0xA5 followed by a write of 0xF1 to this field to unlock the registers. Reading this register returns the current status of lock (0 = Locked with no keys written, 1 = Locked with first key written, 2 = Unlocked). Once unlocked, any write to a lockable register or writing any value to KEY will re-lock the interface.

8.12. PBCFG0 Register Memory Map

Table 8.10. PBCFG0 Memory Map

Register Name	PBCFG0_PBKEY	PBCFG0_XBAR0	PBCFG0_CONTROL1	PBCFG0_CONTROL0
ALL Address	0x4002_A030	0x4002_A020	0x4002_A010	0x4002_A000
Access Methods	ALL	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR
Bit 31	Reserved	XBAR0EN	LOCK	PGDONEF
Bit 30		Reserved	Reserved	Reserved
Bit 29				
Bit 28				PGTIMER
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23				Reserved
Bit 22		AHBEN		
Bit 21		SARADC0TEN		
Bit 20		TMR1EXEN		
Bit 19		TMR1CTEN		
Bit 18		TMR0EXEN		
Bit 17		TMR0CTEN		
Bit 16		CMP1AEN		
Bit 15		CMP1SEN		INT1EN
Bit 14		CMP0AEN		INT1MD
Bit 13		CMP0SEN		
Bit 12		I2C0EN	LPTOSEL	INT1POL
Bit 11		EECI0EN	Reserved	INT1SEL
Bit 10		EPCA0EN		
Bit 9			PMATCHEN	
Bit 8			SPI1SEL	
Bit 7	KEY	SPI0NSSEN	Reserved	INT0EN
Bit 6		SPI0EN		INT0MD
Bit 5		IDAC0TEN		
Bit 4		DMA0T1EN		INT0POL
Bit 3		DMA0T0EN		INT0SEL
Bit 2		USART0CEN	SWVEN	
Bit 1		USART0FCEN	ETMEN	
Bit 0		USART0EN	JTAGEN	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

8.13. PBSTD0, PBSTD1, PBSTD2 and PBSTD3 Registers

This section contains the detailed register descriptions for PBSTD0, PBSTD1, PBSTD2 and PBSTD3 registers.

Register 8.5. PBSTDn_PB: Output Latch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PB															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Addresses

PBSTD0_PB = 0x4002_A0A0
PBSTD1_PB = 0x4002_A140
PBSTD2_PB = 0x4002_A1E0
PBSTD3_PB = 0x4002_A280
This register also supports SET access at (ALL+0x4), CLR access at (ALL+0x8) and MSK access at (ALL+0xC)

Table 8.11. PBSTDn_PB Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PB	**Output Latch.** These bits define the logic level of the port bank output latch. Each bit in this field controls the output latch value for the corresponding port bank pin (bit x controls the latch for pin PBn.x). Digital input pins should be written to 1 and configured for open drain mode. When using this register via the MSK address, the upper 16 bits can be used to mask writes of the lower 16 bits to the corresponding port bank latches.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.6. PBSTDn_PBPIN: Pin Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBPIN															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Addresses

PBSTD0_PBPIN = 0x4002_A0B0
PBSTD1_PBPIN = 0x4002_A150
PBSTD2_PBPIN = 0x4002_A1F0
PBSTD3_PBPIN = 0x4002_A290

Table 8.12. PBSTDn_PBPIN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBPIN	**Pin Value.** These bits read the sensed digital logic level present at the corresponding port bank pin (bit x reads the logic level of pin PBn.x). Pins configured for analog mode will always read back 0.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.7. PBSTDn_PBMDSEL: Mode Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBMDSEL															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Addresses

PBSTD0_PBMDSEL = 0x4002_A0C0
PBSTD1_PBMDSEL = 0x4002_A160
PBSTD2_PBMDSEL = 0x4002_A200
PBSTD3_PBMDSEL = 0x4002_A2A0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.13. PBSTDn_PBMDSEL Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBMDSEL	**Mode Select.** These bits configure the mode of the corresponding port bank pin (bit x controls the mode of pin PBn.x). Setting a bit to 1 configures the pin for digital mode, while clearing a bit to 0 configures the pin for analog mode. Pins configured in analog mode have their digital input paths and weak pullup disconnected.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.8. PBSTDn_PBSKIPEN: Crossbar Pin Skip Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBSKIPEN															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PBSKIPEN = 0x4002_A0D0
PBSTD1_PBSKIPEN = 0x4002_A170
PBSTD2_PBSKIPEN = 0x4002_A210
PBSTD3_PBSKIPEN = 0x4002_A2B0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.14. PBSTDn_PBSKIPEN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBSKIPEN	**Crossbar Pin Skip Enable.** These bits configure the crossbar to skip over the corresponding port bank pin (bit x skips over pin PBn.x). Setting a bit to 1 prevents the crossbar from assigning a peripheral to the pin. Pins which are not available in the current package should be skipped.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.9. PBSTDn_PBOUTMD: Output Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBOUTMD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PBOUTMD = 0x4002_A0E0
PBSTD1_PBOUTMD = 0x4002_A180
PBSTD2_PBOUTMD = 0x4002_A220
PBSTD3_PBOUTMD = 0x4002_A2C0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.15. PBSTDn_PBOUTMD Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBOUTMD	**Output Mode.** These bits configure the digital output mode of the corresponding port bank pin (bit x configures the output mode of pin PBn.x). Setting a bit to 1 configures the pin for push-pull operation. Clearing a bit to 0 configures the pin for open-drain operation. Digital inputs should be configured for open drain mode, with a 1 written to the corresponding output latch.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.10. PBSTDn_PBDRV: Drive Strength

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															PBPUEN
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBDRV															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PBDRV = 0x4002_A0F0
PBSTD1_PBDRV = 0x4002_A190
PBSTD2_PBDRV = 0x4002_A230
PBSTD3_PBDRV = 0x4002_A2D0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.16. PBSTDn_PBDRV Register Bit Descriptions

Bit	Name	Function
31:17	Reserved	Must write reset value.
16	PBPUEN	**Port Bank Weak Pull-up Enable.** Globally enables the weak pull-up for all pins on this port bank. Pins in push-pull or analog mode, or driving a low level in open-drain mode will have their individual weak pullups automatically disabled.
15:0	PBDRV	**Drive Strength.** These bits configure the output drive strength of the corresponding port bank pin (bit x configures the drive strength of pin PBn.x). Setting a bit to 1 enables high drive output on the pin. Clearing a bit to 0 selects low drive output for the pin.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.11. PBSTDn_PM: Port Match Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PM															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PM = 0x4002_A100
PBSTD1_PM = 0x4002_A1A0
PBSTD2_PM = 0x4002_A240
PBSTD3_PM = 0x4002_A2E0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.17. PBSTDn_PM Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PM	**Port Match Value.** When port match is enabled (PMATCHEN = 1), the bits in this register determine the match value for individual pins (bit x configures the match value for pin PBn.x). If the corresponding bit in PMEN is set to 1, a port match event will be triggered when the value in PM matches the logic level at the pin.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.12. PBSTDn_PMEN: Port Match Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PMEN															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PMEN = 0x4002_A110
PBSTD1_PMEN = 0x4002_A1B0
PBSTD2_PMEN = 0x4002_A250
PBSTD3_PMEN = 0x4002_A2F0
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.18. PBSTDn_PMEN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PMEN	**Port Match Enable.** When port match is enabled (PMATCHEN = 1), these bits enable port match on the corresponding port bank pin (bit x enables these functions for pin PBn.x). Setting a bit to 1 enables the pin to be used for port match. Clearing the bit to 0 disables this function for the pin. Pins not supported in the current package should have their match enable bits cleared to 0.

Notes:

1. On SiM3L1x7 devices, PB0 and PB1 consist of 12 pins (PBx.0-PBx.11), PB2 consists of 6 pins (PB2.0-PB2.1, PB2.4-PB2.7), and PB3 is a full port (PB3.0-PB3.15). On SiM3L1x6 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 consists of 11 pins (PB1.0-PB1.10), PB2 consists of 5 pins (PB2.0, PB2.4-PB2.7) and PB3 consists of 12 pins (PB3.0-PB3.11). On SiM3L1x4 devices, PB0 consists of 10 pins (PB0.0-PB0.9), PB1 is not implemented, PB2 consists of 8 pins (PB2.0-PB2.7) and PB3 consists of 10 pins (PB3.0-PB3.9). Any bits in this register controlling unimplemented pins are reserved.

Register 8.13. PBSTDn_PBPGEN: Pulse Generator Pin Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				PBPGEN											
Type	RW				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

PBSTD0_PBPGEN = 0x4002_A120

Table 8.19. PBSTDn_PBPGEN Register Bit Descriptions

Bit	Name	Function
31:12	Reserved	Must write reset value.
11:0	PBPGEN	**Pulse Generator Pin Enable.** These bits enable the pulse generator function on the corresponding port bank pin (bit x enables the pulse generator for pin PBn.x). Setting a bit to 1 enables pulse generation for the pin and clearing a bit to 0 disables pulse generation for the pin.

Register 8.14. PBSTDn_PBPGPHASE: Pulse Generator Phase

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				PBPGPH1											
Type	RW				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				PBPGPH0											
Type	RW				RW											
Reset	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Addresses

PBSTD0_PBPGPHASE = 0x4002_A130

Table 8.20. PBSTDn_PBPGPHASE Register Bit Descriptions

Bit	Name	Function
31:28	Reserved	Must write reset value.
27:16	PBPGPH1	**Pulse Generator Phase 1.** These bits set the logic level for phase 1 of the pulse generator on the corresponding port bank pin (bit x defines phase 1 for pin PBn.x). If pulse generation is enabled on the pin, writing to this register will trigger the beginning of the pulse. The value in PBPGPH0 will be applied to the enabled pins immediately, and when the pulse generator counter times out, the enabled pins will be set to the value in PBPGPH1.
15:12	Reserved	Must write reset value.
11:0	PBPGPH0	**Pulse Generator Phase 0.** These bits set the logic level for phase 0 of the pulse generator on the corresponding port bank pin (bit x defines phase 0 for pin PBn.x). If pulse generation is enabled on the pin, writing to this register will trigger the beginning of the pulse. The value in PBPGPH0 will be applied to the enabled pins immediately, and when the pulse generator counter times out, the enabled pins will be set to the value in PBPGPH1.

8.14. PBSTDn Register Memory Map

Table 8.21. PBSTDn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBSTDn_PB	0x0	ALL | SET | CLR | MASK	Reserved																PB																
PBSTDn_PBPIN	0x10	ALL	Reserved																PBPIN																
PBSTDn_PBMDSEL	0x20	ALL | SET | CLR	Reserved																PBMDSEL																

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: PBSTD0 = 0x4002_A0A0, PBSTD1 = 0x4002_A140, PBSTD2 = 0x4002_A1E0, PBSTD3 = 0x4002_A280

Table 8.21. PBSTDn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBSTDn_PBSKIPEN	0x30	ALL | SET | CLR	Reserved																PBSKIPEN																
PBSTDn_PBOUTMD	0x40	ALL | SET | CLR	Reserved																PBOUTMD																
PBSTDn_PBDRV	0x50	ALL | SET | CLR	Reserved															PBPUEN	PBDRV																
PBSTDn_PM	0x60	ALL | SET | CLR	Reserved																PM																

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: PBSTD0 = 0x4002_A0A0, PBSTD1 = 0x4002_A140, PBSTD2 = 0x4002_A1E0, PBSTD3 = 0x4002_A280

Table 8.21. PBSTDn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBSTDn_PMEN	0x70	ALL | SET | CLR	Reserved																PMEN																
PBSTDn_PBPGEN	0x80	ALL	Reserved																				PBPGEN											
PBSTDn_PBPGPHASE	0x90	ALL	Reserved		PBPGPH1														Reserved		PBPGPH0													

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: PBSTD0 = 0x4002_A0A0, PBSTD1 = 0x4002_A140, PBSTD2 = 0x4002_A1E0, PBSTD3 = 0x4002_A280

8.15. PBGP4 Registers

This section contains the detailed register descriptions for PBGP4 registers.

Register 8.15. PBGP4_PB: Output Latch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PB															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

PBGP4_PB = 0x4002_A320

This register also supports SET access at (ALL+0x4), CLR access at (ALL+0x8) and MSK access at (ALL+0xC)

Table 8.22. PBGP4_PB Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PB	**Output Latch.** These bits define the logic level of the port bank output latch. Each bit in this field controls the output latch value for the corresponding port bank pin (bit x controls the latch for pin PBn.x). Digital input pins should be written to 1 and configured for open drain mode. When using this register via the MSK address, the upper 16 bits can be used to mask writes of the lower 16 bits to the corresponding port bank latches.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.16. PBGP4_PBPIN: Pin Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBPIN															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

PBGP4_PBPIN = 0x4002_A330

Table 8.23. PBGP4_PBPIN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBPIN	**Pin Value.** These bits read the sensed digital logic level present at the corresponding port bank pin (bit x reads the logic level of pin PBn.x). Pins configured for analog mode will always read back 0.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.17. PBGP4_PBMDSEL: Mode Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBMDSEL															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

PBGP4_PBMDSEL = 0x4002_A340

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.24. PBGP4_PBMDSEL Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBMDSEL	**Mode Select.** These bits configure the mode of the corresponding port bank pin (bit x controls the mode of pin PBn.x). Setting a bit to 1 configures the pin for digital mode, while clearing a bit to 0 configures the pin for analog mode. Pins configured in analog mode have their digital input paths and weak pullup disconnected.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.18. PBGP4_PBOUTMD: Output Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBOUTMD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBGP4_PBOUTMD = 0x4002_A350

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.25. PBGP4_PBOUTMD Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PBOUTMD	**Output Mode.** These bits configure the digital output mode of the corresponding port bank pin (bit x configures the output mode of pin PBn.x). Setting a bit to 1 configures the pin for push-pull operation. Clearing a bit to 0 configures the pin for open-drain operation. Digital inputs should be configured for open drain mode, with a 1 written to the corresponding output latch.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.19. PBGP4_PBDRV: Drive Strength

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															PBPUEN
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PBDRV															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBGP4_PBDRV = 0x4002_A360

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.26. PBGP4_PBDRV Register Bit Descriptions

Bit	Name	Function
31:17	Reserved	Must write reset value.
16	PBPUEN	**Port Bank Weak Pull-up Enable.** Globally enables the weak pull-up for all pins on this port bank. Pins in push-pull or analog mode, or driving a low level in open-drain mode will have their individual weak pullups automatically disabled.
15:0	PBDRV	**Drive Strength.** These bits configure the output drive strength of the corresponding port bank pin (bit x configures the drive strength of pin PBn.x). Setting a bit to 1 enables high drive output on the pin. Clearing a bit to 0 selects low drive output for the pin.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.20. PBGP4_PM: Port Match Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PM															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBGP4_PM = 0x4002_A370

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.27. PBGP4_PM Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PM	**Port Match Value.** When port match is enabled (PMATCHEN = 1), the bits in this register determine the match value for individual pins (bit x configures the match value for pin PBn.x). If the corresponding bit in PMEN is set to 1, a port match event will be triggered when the value in PM matches the logic level at the pin.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

Register 8.21. PBGP4_PMEN: Port Match Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PMEN															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PBGP4_PMEN = 0x4002_A380

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 8.28. PBGP4_PMEN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	PMEN	**Port Match Enable.** When port match is enabled (PMATCHEN = 1), these bits enable port match on the corresponding port bank pin (bit x enables these functions for pin PBn.x). Setting a bit to 1 enables the pin to be used for port match. Clearing the bit to 0 disables this function for the pin. Pins not supported in the current package should have their match enable bits cleared to 0.

Notes:

1. On SiM3L1x8-GM and SiM3L1x7-GQ devices PB4 is a full port (PB4.0-PB4.15). On SiM3L1x6 devices PB4 consists of 13 pins (PB4.0-PB4.12). On SiM3L1x4 devices PB4 is not implemented. Any bits in this register controlling unimplemented pins are reserved.

8.16. PBGP4 Register Memory Map

Table 8.29. PBGP4 Memory Map

Register Name	PBGP4_PB	PBGP4_PBPIN	PBGP4_PBMDSEL	PBGP4_PBOUTMD
ALL Address	0x4002_A320	0x4002_A330	0x4002_A340	0x4002_A350
Access Methods	ALL | SET | CLR | MASK	ALL	ALL | SET | CLR	ALL | SET | CLR
Bit 31	Reserved	Reserved	Reserved	Reserved
Bit 30				
Bit 29				
Bit 28				
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23				
Bit 22				
Bit 21				
Bit 20				
Bit 19				
Bit 18				
Bit 17				
Bit 16				
Bit 15	PB	PBPIN	PBMDSEL	PBOUTMD
Bit 14				
Bit 13				
Bit 12				
Bit 11				
Bit 10				
Bit 9				
Bit 8				
Bit 7				
Bit 6				
Bit 5				
Bit 4				
Bit 3				
Bit 2				
Bit 1				
Bit 0				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 8.29. PBGP4 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PBGP4_PBDRV	0x4002_A360	ALL | SET | CLR	Reserved															PBPUEN	PBDRV															
PBGP4_PM	0x4002_A370	ALL | SET | CLR	Reserved																PM															
PBGP4_PMEN	0x4002_A380	ALL | SET | CLR	Reserved																PMEN															

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

9. Power

This section describes the power modes and Power Management Unit of SiM3L1xx devices.

9.1. Power Modes

The SiM3L1xx devices feature seven low power modes in addition to normal operating mode. Several peripherals provide wake up sources for these low power modes, including the Low Power Timer (LPTIMER0), RTC0 (alarms and oscillator failure notification), Comparator 0 (CMP0), Advanced Capture Counter (ACCTR0), LCD VBAT monitor (LCD0), UART0, low power mode charge pump failure, and PMU Pin Wake.

In addition, all peripherals can have their clocks disconnected to reduce power consumption whenever a peripheral is not being used using the clock control (CLKCTRL) registers.

The SiM3L1xx devices have the power modes defined in Table 9.1.

Table 9.1. SiM3L1xx Power Modes

Mode	Description	Mode Entrance	Mode Exit
Normal	■ Core operating at full speed ■ Code executing from flash	Code executing from flash	
Power Mode 1 (PM1)	■ Core operating at full speed ■ Code executing from RAM	Code executing from RAM	
Power Mode 2 (PM2)	■ Core halted ■ AHB, APB and all peripherals operational at full speed	WFI or WFE instruction	Any enabled interrupt source
Power Mode 3 (PM3)	■ All clocks to core and peripherals stopped ■ Faster wake can be enabled by keeping LFOSC0 or RTC0TCLK active	■ DMACTRL0 disabled ■ Fast wake mode: • Fast wake mode enabled in PM3CN • LPOSC0 or RTC0TCLK running and selected as fast wake source • AHB switched to Low Power Oscillator ■ PMSEL cleared to 0 in CLKCTRL0_CONFIG ■ SLEEPDEEP set in the ARM System Control Register ■ WFI or WFE instruction	Requires a wake up source or reset defined by the PMU
Power Mode 4 (PM4)	■ Core operating at low speed ■ Code executing from flash	Code executing from flash	
Power Mode 5 (PM5)	■ Core operating at low speed ■ Code executing from RAM	Code executing from RAM	
Power Mode 6 (PM6)	■ Core halted ■ AHB, APB and all peripherals operational at low speed	WFI or WFE instruction	Any enabled interrupt source

Table 9.1. SiM3L1xx Power Modes

Mode	Description	Mode Entrance	Mode Exit
Power Mode 8 (PM8)	■ Low power sleep ■ The LDO regulators are disabled and all active circuitry operates directly from VBAT ■ The following functions are available: ACCTR0, RTC0, UART0 running from the RTC0TCLK, LPTIMER0, port match, Advanced Capture Counter, and the LCD controller ■ Register state retention	■ PMSEL set to 1 in CLKCTRL0_CONFIG ■ SLEEPDEEP set in the ARM System Control Register ■ WFI or WFE instruction	Requires a wake up source or reset defined by the PMU

9.1.1. Normal Mode (Power Mode 0) and Power Mode 4

Normal Mode and Power Mode 4 are fully operational modes with code executing from flash memory. PM4 is the same as Normal Mode, but with the clocks operating at a lower speed. This enables power to be conserved by reducing the LDO regulator outputs.

9.1.2. Power Mode 1 and Power Mode 5

Power Mode 1 and Power Mode 5 are fully operational modes with code executing from RAM. PM5 is the same as PM1, but with the clocks operating at a lower speed. This enables power to be conserved by reducing the LDO regulator outputs. Compared with the corresponding flash operational mode (Normal or PM4), the active power consumption of the device in these modes is reduced. Additionally, at higher speeds in PM1, the core throughput can also be increased because RAM doesnot require additional wait states that reduce the instruction fetch speed.

9.1.3. Power Mode 2 and Power Mode 6

In Power Mode 2 and Power Mode 6, the core halts and the peripherals continue to run at the selected clock speed. PM6 is the same as PM2, but with the clocks operating at a lower speed. This enables power to be conserved by reducing the LDO regulator outputs. To place the device in PM2 or PM6, the core should execute a wait-for-interrupt (WFI) or wait-for-event (WFE) instruction. If the WFI instruction is called from an interrupt service routine, the interrupt that wakes the device from PM2 or PM6 must be of a sufficient priority to be recognized by the core. It is recommended to perform both a DSB (Data Synchronization Barrier) and an ISB (Instruction Syncronization Barrier) operation prior to the WFI to ensure all bus accesses complete. When operating from the LFOSC0, PM6 can achieve similar power consumption to PM3, but with faster wake times and the ability to wake on any interrupt.

9.1.4. Power Mode 3

In Power Mode 3 the core and peripheral clocks are halted. The available sources to wake from PM3 are controlled by the Power Management Unit (PMU). A special Fast Wake option allows the core to wake faster by keeping the LFOSC0 or RTC0 clock active. Because the current consumption of these blocks is minimal, it is recommended to use the fast wake option.

Before entering PM3, the DMA controller should be disabled, and the desired wake source(s) should be configured in the PMU. The SLEEPDEEP bit in the ARM System Control Register should be set, and the PMSEL bit in the CLKCTRL0_CONFIG register should be cleared to indicate that PM3 is the desired power mode. For fast wake, the core clocks (AHB and APB) should be configured to run from the LPOSC, and the PM3 Fast wake option and clock source should be selected in the PM3CN register.

The device will enter PM3 on a WFI or WFE instruction. If the WFI instruction is called from an interrupt service routine, the interrupt that wakes the device from PM3 must be of a sufficient priority to be recognized by the core. It is recommended to perform both a DSB (Data Synchronization Barrier) and an ISB (Instruction Synchronization Barrier) operation prior to the WFI to ensure all bus access is complete.

9.1.5. Power Mode 8

In Power Mode 8, the core and most peripherals are completely powered down, but all registers and selected RAM blocks retain their state. The LDO regulators are disabled, so all active circuitry operates directly from VBAT. Alternatively, the PMU has a specialized VBAT-divided-by-2 charge pump that can power some internal modules while in PM8 to save power. The fully operational functions in this mode are: LPTIMER0 , RTC0, UART0 running from RTC0TCLK, PMU Pin Wake, the advanced capture counter, and the LCD controller.

This mode provides the lowest power consumption for the device, but requires an appropriate wake up source or reset to exit. The available wake up or reset sources to wake from PM8 are controlled by the Power Management Unit (PMU). The available wake up sources are: Low Power Timer (LPTIMER0), RTC0 (alarms and oscillator failure notification), Comparator 0 (CMP0), advanced capture counter (ACCTR0), LCD VBAT monitor (LCD0), UART0, low power mode charge pump failure, and PMU Pin Wake. The available reset sources are: RESET pin, VBAT supply monitor, Comparator 0, Comparator 1, low power mode charge pump failure, RTC0 oscillator failure, or a PMU wake event.

Before entering PM8, the desired wake source(s) should be configured in the PMU. The SLEEPDEEP bit in the ARM System Control Register should be set, and the PMSEL bit in the CLKCTRL0_CONFIG register should be set to indicate that PM9 is the desired power mode.

The device will enter PM8 on a WFI or WFE instruction, and remain in PM9 until a reset configured by the PMU occurs. It is recommended to perform both a DSB (Data Synchronization Barrier) and an ISB (Instruction Synchronization Barrier) operation prior to the WFI to ensure all bus access is complete.

To ensure the lowest possible power consumption in PM8, firmware should take the following steps:

1. Set the RAM retention enable bits PMU0_CONTROL register to enable the blocks of retention RAM required for the application. The RAM block containing the stack should be included to prevent a stack error on wake. Enable all RAM blocks if the stack location or RAM usage is in uncertain.
2. If using the low-power charge pump, configure the enable and interrupt in the PMU0_CONTROL register.
3. Enable the PMSEL bit in the CLKCTRL0_CONFIG register.
4. Configure the desired interrupt/wake sources in their respective peripheral interfaces.
5. Enable the desired wake source(s) using the PMU0_WAKEEN register. Clear all undesired wake sources.
6. If the desired wake source is a pin wake, set the PWAKEEN bit in the PMU0_CONTROL register.
7. Clear the corresponding reset source bits in the RSTSRC0_RESETEN register. If the respective reset enable bit is set in the RSTSRC0_RESETEN, an event will generate a reset not a wake.
8. Clear pending interrupt requests for the desired wake sources and then enable the interrupt sources in the NVIC. A wake event will occur only if the interrupt is enabled.
9. Enable global interrupt requests.
10. Ensure that the corresponding interrupt has a valid interrupt handler. Code execution will jump to the interrupt handler on a wake event. The core will generate an exception on wake if no handler exists.
11. Stop the watchdog timer if it is running.
12. Turn off the systick timer.
13. Clear all wakeup flags by setting the WAKECLR bit in the PMU0_CONTROL register. The MCU will not sleep if these flags are set.
14. Clear the PMU0_STATUS register. (Clears PM8EF, PWAKEF, and PORF.) The MCU will not sleep if these flags are set.
15. Set the SLEEPDEEP bit in the ARM System Control Register to 1.
16. **Configure the LDO output voltages to the recommended settings for the external supply range**. If adjusting the LDOs to a higher voltage, change the analog LDO first. If adjusting to a lower voltage, change the analog LDO last.
17. Execute a DSB and an ISB operation.
18. Execute a WFI instruction.
19. The core will remain in PM8 until an enabled wake up or a reset event occurs.

20. On a wake event, code execution will jump to the interrupt handler, then resume from the WFI instruction.
21. Validate that PM8EF bit in the PMU0_STATUS register is set.
22. Read the last wake source from the PMU0_WAKESTATUS register.
23. **Configure the LDO output voltages to the desired operational output level**. If adjusting the LDOs to a higher voltage, change the analog LDO first. If adjusting to a lower voltage, change the analog LDO last.
24. Handle all enabled wake events.
25. **Note that the WDTIMER and PVTOSC peripherals are both reset when operating in PM8.** For example, if the WDTIMER was disabled when entering PM8, it will be enabled (default state) upon exit from PM8. These peripherals should be handled accordingly in firmware.

10. Power Management Unit (PMU0)

This section describes the Power Management Unit (PMU) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

The PMU module includes the following features:

- Provides the enable or disable for the analog power system, including the three LDO regulators.
- Manages the source for the VDRV output when entering and leaving Power Mode 8.
- Up to 14 pin wake inputs can wake the device from Power Mode 8.
- The Low Power Timer, RTC0 (alarms and oscillator failure), Comparator 0, pulse counter, LCD0 VBAT monitor, UART0, low power mode charge pump failure, and the RESET pin can also serve as wake sources for Power Mode 8.
- All PM8 wake sources (except for the RESET pin) can also reset the Low Power Timer or RTC0 modules.
- Controls which 4 kB RAM blocks are retained while in Power Mode 8.
- Provides a PMU_Asleep signal to a pin as an indicator that the device is in PM8.
- Specialized charge pump to reduce power consumption in PM8.
- Provides control for the internal switch between VBAT and VDC to power the VDRV pin for external circuitry.

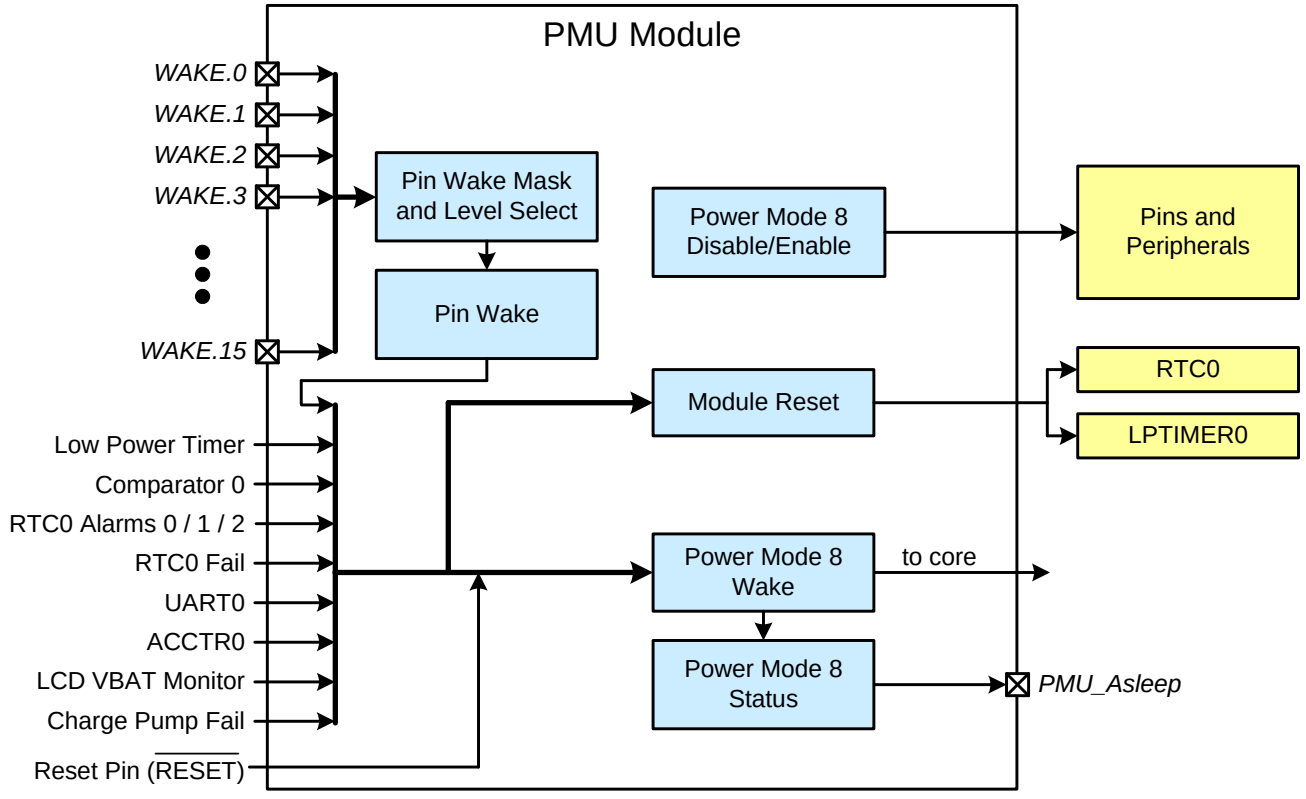


Figure 10.1. PMU Block Diagram

The PMU manages the power-up sequence on power on and the wake up sources from PM8. On power-up, the PMU ensures the core voltages are of a proper value before core instruction execution begins.

10.1. Waking from Power Mode 8

The wake signals for power mode 8 can be sourced from pins (Pin Wake), the Low Power Timer, UART0, Comparator 0, RTC0 Alarms (0, 1, or 2), RTC0 Fail, ACCTR0, the LCD VBAT monitor or charge pump supply, or a pin reset. In most cases, the corresponding interrupt enable must be set in the module in order for an event to be a wakeup source. The Comparator module is the exception and the wakeup event will occur even if the interrupt is disabled. These wakeup sources can also be optionally used to reset RTC0 or the Low Power Timer while the device remains in PM8.

Because resetting the device is a valid exit from PM8, certain applications may want to know upon reset if the device was previously configured in PM8, Firmware can check the PM8EF bit during the initialization sequence to determine if the device reset while in PM8, and act accordingly. If the device did reset while in PM8, firmware must clear the bits keeping the peripheral and pin interfaces in a lower power state (PERILPEN and PINLPEN), and the WAKESTATUS register provides status flags to indicate the wakeup source. The WAKESTATUS register can be cleared by writing 0 to the WAKECLR bit.

10.1.1. Pin Wake

The available Pin Wake sources (WAKE.0-WAKE.15) can be used to wake the device from low power sleep modes. The pin wake function can be enabled for a pin by setting the corresponding bit in the PWEN register, and setting the PWAKEEN bit to 1. The desired polarity of the pin to wake the device should be programmed into the corresponding bit in the PWPOL register. To serve as a wake source, a pin must remain active in the matching state long enough for the PMU to detect the wake up signal (50ns). The pin wake trigger sources vary by package and are shown in Table 10.1.

Table 10.1. Pin Wake Sources

Pin Wake Source	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
WAKE.0	PB0.0	PB0.0	PB0.0
WAKE.1	PB0.1	Reserved	Reserved
WAKE.2	PB0.2	PB0.1	PB0.1
WAKE.3	PB0.3	PB0.2	PB0.2
WAKE.4	PB0.4	PB0.3	PB0.3
WAKE.5	PB0.5	PB0.4	PB0.4
WAKE.6	PB0.6	PB0.5	PB0.5
WAKE.7	PB0.7	PB0.6	Reserved
WAKE.8	PB0.8	PB0.7	PB0.6
WAKE.9	PB0.9	PB0.8	Reserved
WAKE.10	PB0.10	PB0.9	Reserved
WAKE.11	PB0.11	Reserved	Reserved
WAKE.12	PB2.0	PB2.0	PB2.0

Table 10.1. Pin Wake Sources

Pin Wake Source	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
WAKE.13	PB2.1	Reserved	PB2.1
WAKE.14	Reserved	Reserved	PB2.2
WAKE.15	Reserved	Reserved	PB2.3

10.1.2. Low Power Timer

The Low Power Timer wake up source is caused by a Low Power Timer overflow.

10.1.3. Comparator 0

A Comparator 0 (CMP0) event can serve as a wake up or reset source in the PMU. This event can occur from a rising, falling, or either edge on the Comparator 0 output, depending on the settings in the Comparator module.

10.1.4. RTC0

The RTC0 Alarms (0, 1, and 2) and Fail events are wake up sources from PM8 or can automatically reset the LPTIMER0 and RTC0 modules. The Alarms occur from a match between the RTC0 timer and the corresponding Alarm compare value. The RTC0 Oscillator Fail event occurs when the RTC0 Missing Clock Detector is enabled and the RTC0OSC falls below the missing clock detector trigger frequency.

10.1.5. UART0

UART0 can operate while the device is in PM8. Enabling UART0 as a wake source allows the device to wake on UART0 activity.

10.1.6. ACCTR0

An ACCTR0 event can also serve as a wakeup source for the device. The ACCTR0 block can be configured to wake the device to process data after a certain number of events have occurred.

10.1.7. LCD VBAT Supply Monitor

When the LCD VBAT supply is enabled as a wakeup source, the device will exit PM8 when the VBAT supply drops below the defined threshold.

10.1.8. Charge Pump Supply Fail

When enabled, a charge pump supply fail event can also wake the device from PM8.

10.2. Retention RAM Control

On-chip RAM is segmented into 4 kB blocks which can be individually selected to retain state in Power Mode 8. The control bits for each 4 kB block are RAM0REN through RAM7REN in the CONTROL register. Setting one of these bits to 1 will enable retention of that RAM block. RAM blocks which do not have retention enabled will lose their contents if the device enters PM8. Each block selected to retain state will consume a small amount of extra current, so it is advisable to structure the application's memory usage such that the minimal amount of retention RAM is required.

Note: To ensure robust operation at wakeup, it is important to understand the location of any RAM variables that are required to maintain state. This includes global and local static variables, as well as the placement of the stack.

10.3. PMU0 Registers

This section contains the detailed register descriptions for PMU0 registers.

Register 10.1. PMU0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								RAM7REN	RAM6REN	RAM5REN	RAM4REN	RAM3REN	RAM2REN	RAM1REN	RAM0REN
Type	R								RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									CPMONIEN	CPMONEN	PMUASLPEN	PWAKEEN	Reserved		WAKECLR
Type	R		RW				R			RW	RW	RW	RW	R		W
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

Register ALL Access Address

PMU0_CONTROL = 0x4004_8000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.2. PMU0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23	RAM7REN	**RAM 7 Retention Enable.** When set to 1, the RAM 7 block is powered (4 kB addresses from 0x20007000 to 0x20007FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
22	RAM6REN	**RAM 6 Retention Enable.** When set to 1, the RAM 6 block is powered (4 kB addresses from 0x20006000 to 0x20006FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
21	RAM5REN	**RAM 5 Retention Enable.** When set to 1, the RAM 5 block is powered (4 kB addresses from 0x20005000 to 0x20005FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.

Table 10.2. PMU0_CONTROL Register Bit Descriptions

Bit	Name	Function
20	RAM4REN	**RAM 4 Retention Enable.** When set to 1, the RAM 4 block is powered (4 kB addresses from 0x20004000 to 0x20004FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
19	RAM3REN	**RAM 3 Retention Enable.** When set to 1, the RAM 3 block is powered (4 kB addresses from 0x20003000 to 0x20003FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
18	RAM2REN	**RAM 2 Retention Enable.** When set to 1, the RAM 2 block is powered (4 kB addresses from 0x20002000 to 0x20002FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
17	RAM1REN	**RAM 1 Retention Enable.** When set to 1, the RAM 1 block is powered (4 kB addresses from 0x20001000 to 0x20001FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
16	RAM0REN	**RAM 0 Retention Enable.** When set to 1, the RAM 0 block is powered (4 kB addresses from 0x20000000 to 0x20000FFF). This bit should be cleared to 0 when entering Power Mode 8 to save power if these RAM locations do not need to be retained.
15:7	Reserved	Must write reset value.
6	CPMONIEN	**Low Power Charge Pump Voltage Monitor Interrupt Enable.** 0: Disable the low power charge pump voltage monitor interrupt. 1: Enable the low power charge pump voltage monitor interrupt.
5	CPMONEN	**Low Power Charge Pump Voltage Monitor Enable.** 0: Disable the low power charge pump voltage monitor. 1: Enable the low power charge pump voltage monitor.
4	PMUASLPEN	**PMU Asleep Pin Enable.** When set to 1, the PMU Asleep signal will be sent to the appropriate pin. This pin should be skipped by the Crossbar if firmware enables the PMU Asleep signal.
3	PWAKEEN	**Pin Wake Match Enable.** 0: Disable Pin Wake. 1: Enable Pin Wake.
2:1	Reserved	Must write reset value.
0	WAKECLR	**Wakeup Source Clear.** Writing a 0 to this bit clears all wakeup sources. 0: Clear all wakeup sources. 1: Reserved.

Register 10.2. PMU0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				CPLOAD		Reserved	CPEN	Reserved	VDRVSMD		Reserved				
Type	RW				RW		RW	RW	R	RW		RW				
Reset	X	X	X	X	0	0	0	0	0	0	0	0	X	X	X	X

Register ALL Access Address

PMU0_CONFIG = 0x4004_8010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.3. PMU0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:12	Reserved	Must write reset value.
11:10	CPLOAD	**Charge Pump Load Setting.**
9	Reserved	Must write reset value.
8	CPEN	**Low Power Charge Pump Enable.**
7	Reserved	Must write reset value.
6:5	VDRVSMD	**VDRV Switch Mode.** 00: High-Z. 01: Reserved. 10: VBAT connected to VDRV. 11: DC-DC output connected to VDRV.
4:0	Reserved	Must write reset value.

Register 10.3. PMU0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved												CPSTS	PORF	PWAKEF	PM8EF
Type	R												R	RW	R	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

Register ALL Access Address

PMU0_STATUS = 0x4004_8020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.4. PMU0_STATUS Register Bit Descriptions

Bit	Name	Function
31:4	Reserved	Must write reset value.
3	CPSTS	**Low Power Charge Pump Voltage Monitor Status.** 0: The low power charge pump supply voltage is below the threshold. 1: The low power charge pump supply voltage is greater than the threshold.
2	PORF	**Power-On Reset Flag.** Hardware sets this bit to 1 to indicate that a power-on reset event occurred. This bit must be cleared by firmware.
1	PWAKEF	**Pin Wake Status Flag.** When set to 1, this flag indicates that pin wake condition is active.
0	PM8EF	**Power Mode 8 Exited Flag.** When set to 1, this flag indicates that the device exited Power Mode 8. Firmware must clear this flag.

Register 10.4. PMU0_WAKEEN: Wakeup Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved							CPFWEN	UART0WEN	LPT0WEN	PWAKEWEN	LCDMONWEN	ACC0WEN	CMP0WEN	RTC0A0WEN	RTC0FWEN
Type	R							RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PMU0_WAKEEN = 0x4004_8030

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.5. PMU0_WAKEEN Register Bit Descriptions

Bit	Name	Function
31:9	Reserved	Must write reset value.
8	CPFWEN	**Low Power Charge Pump Supply Fail Wake Enable.** When set to 1, a low power charge pump supply fail event will wake the device from Power Mode 8.
7	UART0WEN	**UART0 Wake Enable.** When set to 1, a UART0 event will wake the device from Power Mode 8.
6	LPT0WEN	**Low Power Timer Wake Enable.** When set to 1, an LPTIMER0 event will wake the device from Power Mode 8.
5	PWAKEWEN	**Pin Wake Wake Enable.** When set to 1, a Pin Wake event will wake the device from Power Mode 8.
4	LCDMONWEN	**LCD VBAT Voltage Monitor Wake Enable.** When set to 1, an LCD VBAT voltage monitor event will wake the device from Power Mode 8.
3	ACC0WEN	**Advanced Capture Counter 0 Wake Enable.** When set to 1, an Advanced Capture Counter (ACCTR0) event will wake the device from Power Mode 8.
2	CMP0WEN	**Comparator 0 Wake Enable.** When set to 1, a Comparator 0 event will wake the device from Power Mode 8.

Table 10.5. PMU0_WAKEEN Register Bit Descriptions

Bit	Name	Function
1	RTC0A0WEN	**RTC0 Alarm Wake Enable.** When set to 1, an RTC0 Alarm event will wake the device from Power Mode 8.
0	RTC0FWEN	**RTC0 Fail Wake Enable.** When set to 1, an RTC0 Fail event will wake the device from Power Mode 8.

Register 10.5. PMU0_WAKESTATUS: Wakeup Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						RSTWF	CPFWF	UART0WF	LPT0WF	PWAKEWF	LCDMONWF	ACC0WF	CMP0WF	RTC0A0WF	RTC0FWF
Type	R						R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

PMU0_WAKESTATUS = 0x4004_8040

Table 10.6. PMU0_WAKESTATUS Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	RSTWF	**Reset Pin Wake Flag.** When set to 1, this flag indicates that the $\overline{\text{RESET}}$ pin woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
8	CPFWF	**Low Power Charge Pump Supply Fail Wake Flag.** When set to 1, this flag indicates that a low power charge pump supply fail event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
7	UART0WF	**UART0 Wake Flag.** When set to 1, this flag indicates that a UART0 event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
6	LPT0WF	**Low Power Timer Wake Flag.** When set to 1, this flag indicates that a LPTIMER0 event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
5	PWAKEWF	**Pin Wake Wake Flag.** When set to 1, this flag indicates that a Pin Wake event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
4	LCDMONWF	**LCD VBAT Voltage Monitor Wake Flag.** When set to 1, this flag indicates that a LCD VBAT voltage monitor event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.

Table 10.6. PMU0_WAKESTATUS Register Bit Descriptions

Bit	Name	Function
3	ACC0WF	**Advanced Capture Counter 0 Wake Flag.** When set to 1, this flag indicates that an Advanced Capture Counter (ACCTR0) event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
2	CMP0WF	**Comparator 0 Wake Flag.** When set to 1, this flag indicates that a Comparator 0 event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
1	RTC0A0WF	**RTC0 Alarm Wake Flag.** When set to 1, this flag indicates that an RTC0 Alarm event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.
0	RTC0FWF	**RTC0 Fail Wake Flag.** When set to 1, this flag indicates that an RTC0 fail event woke the device. Firmware must clear this flag using the WAKECLR bit in the CONTROL register.

Register 10.6. PMU0_PWEN: Pin Wake Pin Enable

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PW15EN	PW14EN	PW13EN	PW12EN	PW11EN	PW10EN	PW9EN	PW8EN	PW7EN	PW6EN	PW5EN	PW4EN	PW3EN	PW2EN	PW1EN	PW0EN
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PMU0_PWEN = 0x4004_8050

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.7. PMU0_PWEN Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15	PW15EN	**WAKE.15 Enable.** When set to 1, this bit enables the WAKE.15 signal to wake the device.
14	PW14EN	**WAKE.14 Enable.** When set to 1, this bit enables the WAKE.14 signal to wake the device.
13	PW13EN	**WAKE.13 Enable.** When set to 1, this bit enables the WAKE.13 signal to wake the device.
12	PW12EN	**WAKE.12 Enable.** When set to 1, this bit enables the WAKE.12 signal to wake the device.
11	PW11EN	**WAKE.11 Enable.** When set to 1, this bit enables the WAKE.11 signal to wake the device.
10	PW10EN	**WAKE.10 Enable.** When set to 1, this bit enables the WAKE.10 signal to wake the device.
9	PW9EN	**WAKE.9 Enable.** When set to 1, this bit enables the WAKE.9 signal to wake the device.
8	PW8EN	**WAKE.8 Enable.** When set to 1, this bit enables the WAKE.8 signal to wake the device.
7	PW7EN	**WAKE.7 Enable.** When set to 1, this bit enables the WAKE.7 signal to wake the device.

Table 10.7. PMU0_PWEN Register Bit Descriptions

Bit	Name	Function
6	PW6EN	**WAKE.6 Enable.** When set to 1, this bit enables the WAKE.6 signal to wake the device.
5	PW5EN	**WAKE.5 Enable.** When set to 1, this bit enables the WAKE.5 signal to wake the device.
4	PW4EN	**WAKE.4 Enable.** When set to 1, this bit enables the WAKE.4 signal to wake the device.
3	PW3EN	**WAKE.3 Enable.** When set to 1, this bit enables the WAKE.3 signal to wake the device.
2	PW2EN	**WAKE.2 Enable.** When set to 1, this bit enables the WAKE.2 signal to wake the device.
1	PW1EN	**WAKE.1 Enable.** When set to 1, this bit enables the WAKE.1 signal to wake the device.
0	PW0EN	**WAKE.0 Enable.** When set to 1, this bit enables the WAKE.0 signal to wake the device.

Register 10.7. PMU0_PWPOL: Pin Wake Pin Polarity Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PW15POL	PW14POL	PW13POL	PW12POL	PW11POL	PW10POL	PW9POL	PW8POL	PW7POL	PW6POL	PW5POL	PW4POL	PW3POL	PW2POL	PW1POL	PW0POL
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PMU0_PWPOL = 0x4004_8060

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 10.8. PMU0_PWPOL Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15	PW15POL	**WAKE.15 Polarity Select.** If PW15EN is set, this bit selects the logic level of WAKE.15 that will cause a Pin Wake event.
14	PW14POL	**WAKE.14 Polarity Select.** If PW14EN is set, this bit selects the logic level of WAKE.14 that will cause a Pin Wake event.
13	PW13POL	**WAKE.13 Polarity Select.** If PW13EN is set, this bit selects the logic level of WAKE.13 that will cause a Pin Wake event.
12	PW12POL	**WAKE.12 Polarity Select.** If PW12EN is set, this bit selects the logic level of WAKE.12 that will cause a Pin Wake event.
11	PW11POL	**WAKE.11 Polarity Select.** If PW11EN is set, this bit selects the logic level of WAKE.11 that will cause a Pin Wake event.
10	PW10POL	**WAKE.10 Polarity Select.** If PW10EN is set, this bit selects the logic level of WAKE.10 that will cause a Pin Wake event.

Table 10.8. PMU0_PWPOL Register Bit Descriptions

Bit	Name	Function
9	PW9POL	**WAKE.9 Polarity Select.** If PW9EN is set, this bit selects the logic level of WAKE.9 that will cause a Pin Wake event.
8	PW8POL	**WAKE.8 Polarity Select.** If PW8EN is set, this bit selects the logic level of WAKE.8 that will cause a Pin Wake event.
7	PW7POL	**WAKE.7 Polarity Select.** If PW7EN is set, this bit selects the logic level of WAKE.7 that will cause a Pin Wake event.
6	PW6POL	**WAKE.6 Polarity Select.** If PW6EN is set, this bit selects the logic level of WAKE.6 that will cause a Pin Wake event.
5	PW5POL	**WAKE.5 Polarity Select.** If PW5EN is set, this bit selects the logic level of WAKE.5 that will cause a Pin Wake event.
4	PW4POL	**WAKE.4 Polarity Select.** If PW4EN is set, this bit selects the logic level of WAKE.4 that will cause a Pin Wake event.
3	PW3POL	**WAKE.3 Polarity Select.** If PW3EN is set, this bit selects the logic level of WAKE.3 that will cause a Pin Wake event.
2	PW2POL	**WAKE.2 Polarity Select.** If PW2EN is set, this bit selects the logic level of WAKE.2 that will cause a Pin Wake event.
1	PW1POL	**WAKE.1 Polarity Select.** If PW1EN is set, this bit selects the logic level of WAKE.1 that will cause a Pin Wake event.
0	PW0POL	**WAKE.0 Polarity Select.** If PW0EN is set, this bit selects the logic level of WAKE.0 that will cause a Pin Wake event.

10.4. PMU0 Register Memory Map

Table 10.9. PMU0 Memory Map

Register Name	PMU0_WAKEEN	PMU0_STATUS	PMU0_CONFIG	PMU0_CONTROL
ALL Address	0x4004_8030	0x4004_8020	0x4004_8010	0x4004_8000
Access Methods	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR
Bit 31	Reserved	Reserved	Reserved	Reserved
Bit 30				
Bit 29				
Bit 28				
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23				RAM7REN
Bit 22				RAM6REN
Bit 21				RAM5REN
Bit 20				RAM4REN
Bit 19				RAM3REN
Bit 18				RAM2REN
Bit 17				RAM1REN
Bit 16				RAM0REN
Bit 15				Reserved
Bit 14				
Bit 13				
Bit 12				
Bit 11			CPLOAD	
Bit 10			Reserved	
Bit 9			CPEN	
Bit 8			Reserved	
Bit 7			VDRVSMD	
Bit 6			Reserved	CPMONIEN
Bit 5	CPFWEN			CPMONEN
Bit 4	UART0WEN			PMUASLPEN
Bit 3	LPT0WEN	CPSTS		PWAKEEN
Bit 2	PWAKEWEN	PORF		Reserved
Bit 1	LCDMONWEN	PWAKEF		
Bit 0	ACC0WEN	PM8EF		WAKECLR
	CMP0WEN			
	RTC0A0WEN			
	RTC0FWEN			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 10.9. PMU0 Memory Map

Register Name	PMU0_WAKESTATUS	PMU0_PWEN	PMU0_PWPOL
ALL Address	0x4004_8040	0x4004_8050	0x4004_8060
Access Methods	ALL	ALL | SET | CLR	ALL | SET | CLR
Bit 31	Reserved	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15		PW15EN	PW15POL
Bit 14		PW14EN	PW14POL
Bit 13		PW13EN	PW13POL
Bit 12		PW12EN	PW12POL
Bit 11		PW11EN	PW11POL
Bit 10		PW10EN	PW10POL
Bit 9	RSTWF	PW9EN	PW9POL
Bit 8	CPFWF	PW8EN	PW8POL
Bit 7	UART0WF	PW7EN	PW7POL
Bit 6	LPT0WF	PW6EN	PW6POL
Bit 5	PWAKEWF	PW5EN	PW5POL
Bit 4	LCDMONWF	PW4EN	PW4POL
Bit 3	ACC0WF	PW3EN	PW3POL
Bit 2	CMP0WF	PW2EN	PW2POL
Bit 1	RTC0A0WF	PW1EN	PW1POL
Bit 0	RTC0FWF	PW0EN	PW0POL

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

11. Internal Voltage Regulator (LDO0)

This section describes the internal voltage regulator (LDO) block, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

11.1. Internal Voltage Regulator Features

The internal voltage regulator block includes the following features:

- Three independent regulators for the digital, analog and memory subsystems.
- LDOs are adjustable to allow power savings.
- Inputs are selectable between the VBAT pin and VDC pin (dc-dc output).
- High and low bias configurations for each regulator to save power.

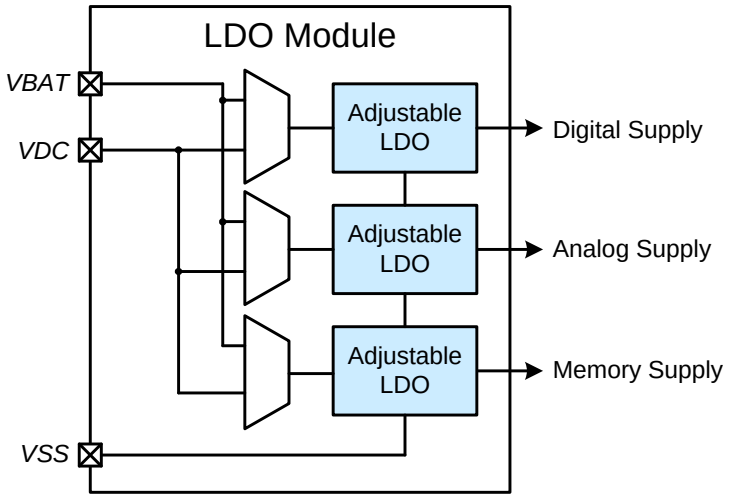


Figure 11.1. LDO0 Block Diagram

11.2. Functional Description

The core voltage regulator consists of three independent LDO regulators for the memory, analog and digital subsystems. Each regulator can be powered from either the VBAT pin or the DCDC converter output at the VDC pin, and has an adjustable bias setting for power savings. Additionally, the output of the regulators are adjustable, and can be used to scale the power consumption back in certain applications. The CONTROL register allows firmware to program the regulator properties as needed.

11.2.1. Input Source Selection

By default, the input to all three LDOs is the VBAT supply pin. In many cases, the DCDC converter can be used to realize additional power efficiency for the device. To operate any of the LDOs from the DCDC converter output, firmware must first configure and enable the DCDC converter. When running the LDOs from the DCDC converter, it is recommended to set the DCDC output at least 100 mV above the highest LDO output, to ensure enough headroom. After the DCDC is configured and running, the LDOs can be individually switched to run from the DCDC output using the DLDOSSEL, MLDOSSEL and ALDOSSEL bits in the CONTROL register. As long as the voltage at VDC and VBAT are stable and within the device supply range, the LDOs can be switched seamlessly between the two power sources at any time.

11.2.2. Bias Current Configuration

Each LDO has configuration options to select high or low bias current. The bias current affects the response time of the LDO. High bias current is useful in situations where the load current on the LDO may rise very quickly—when switching from a slow clock source to a fast clock source for example. In general, low bias may be used any time the device is operated at constant loads, or if the load will not instantaneously change more than a few mA. The bits DLDOBSEL, MLDOBSEL and ALDOBSEL control the bias setting of the individual LDOs.

11.2.3. Adjustable Output

The LDO outputs are adjustable, with a valid range of 0.8 to 1.9 V. Additional power savings for the device may be realized by lowering the LDO output voltage of the memory and digital regulators under appropriate conditions, such as lower clock speeds. The regulators are adjusted using the ALDOOVAL, DLDOOVAL and MLDOOVAL bit fields in the CONTROL register. The adjustment fields are linear, with each LSB representing approximately 50 mV. To adjust the memory LDO to 1.6 V, for example, the MLDOOVAL field can be set to 0x10 (0.8 V + 50 mV x 16). The LDO outputs are also available to the SARADC and Comparators for monitoring purposes.

The optimal LDO settings for different operational speeds are to be determined (TBD), pending full characterization of silicon over process, temperature, and voltage corners. Preliminary silicon characterization suggests that the safe minimum voltage for the memory LDO at all speeds is 1.6 V. The digital LDO may be safely set to 1.0 V at speeds of 20 MHz or less, and 1.3 V between 20 and 50 MHz. The analog LDO should normally be left at 1.8 V output during active operation. All LDOs may be adjusted prior to entering PM8, to extend the external supply operating range. See the electrical specification tables in the data sheet for specified output settings.

11.2.3.1. Important Notes About Adjustable LDOs

1. The analog LDO output should always be set equal to or higher than the output of the memory LDO. When lowering both LDOs (for example to go into PM8 under low supply conditions), first adjust the memory LDO and then the analog LDO. When raising the output of both LDOs, adjust the analog LDO before adjusting the memory LDO.
2. Before entering PM8, all three LDOs should be adjusted to the recommended setting for the external supply range, according to the data sheet specifications.
3. When writing to or erasing flash memory, it setting the output of the memory LDO to 1.8 V or higher is required.

11.3. LDO0 Registers

This section contains the detailed register descriptions for LDO0 registers.

Register 11.1. LDO0_CONTROL: Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved									DLDOSSEL	DLDOBSEL	DLDOOVAL				
Type	R						RW		R	RW	RW	RW				
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	MLDOSSEL	MLDOBSEL	MLDOOVAL					Reserved	ALDOSSEL	ALDOBSEL	ALDOOVAL				
Type	R	RW	RW	RW					R	RW	RW	RW				
Reset	0	0	1	1	0	1	0	0	0	0	1	1	0	1	0	0

Register ALL Access Address

LDO0_CONTROL = 0x4003_9000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 11.1. LDO0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:23	Reserved	Must write reset value.
22	DLDOSSEL	**Digital LDO Source Select.** 0: Select the VBAT pin as the input voltage to the digital LDO. 1: Select the output of the DC-DC converter as the input voltage to the digital LDO.
21	DLDOBSEL	**Digital LDO Bias Select.** 0: Select a low bias for the digital LDO. 1: Select a high bias for the digital LDO.
20:16	DLDOOVAL	**Digital LDO Output Value Select.** This field configures the output voltage of the digital LDO between 0.8 and 1.9 V in 50 mV steps. The reset value of this field is 1.8 V.
15	Reserved	Must write reset value.
14	MLDOSSEL	**Memory LDO Source Select.** 0: Select the VBAT pin as the input voltage to the memory LDO. 1: Select the output of the DC-DC converter as the input voltage to the memory LDO.

Table 11.1. LDO0_CONTROL Register Bit Descriptions

Bit	Name	Function
13	MLDOBSEL	**Memory LDO Bias Select.** 0: Select a low bias for the memory LDO. 1: Select a high bias for the memory LDO.
12:8	MLDOOVAL	**Memory LDO Output Value Select.** This field configures the output voltage of the memory LDO between 0.8 and 1.9 V in 50 mV steps. The reset value of this field is 1.8 V.
7	Reserved	Must write reset value.
6	ALDOSSEL	**Analog LDO Source Select.** 0: Select the VBAT pin as the input voltage to the analog LDO. 1: Select the output of the DC-DC converter as the input voltage to the analog LDO.
5	ALDOBSEL	**Analog LDO Bias Select.** 0: Select a low bias for the analog LDO. 1: Select a high bias for the analog LDO.
4:0	ALDOOVAL	**Analog LDO Output Value Select.** This field configures the output voltage of the analog LDO between 0.8 and 1.9 V in 50 mV steps. The reset value of this field is 1.8 V.

11.4. LDO0 Register Memory Map

Table 11.2. LDO0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
LDO0_CONTROL	0x4003_9000	ALL | SET | CLR	Reserved									DLDOSSEL	DLDOBSEL	DLDOOVAL					Reserved	MLDOSSEL	MLDOBSEL	MLDOOVAL				Reserved	ALDOSSEL	ALDOBSEL	ALDOOVAL					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

12. DC-DC Regulator (DCDC0)

This section describes the dc-dc Regulator (DCDC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the DCDC block, which is used by all device families covered in this document.

12.1. DCDC Features

The DCDC module includes the following features:

- Efficiently utilizes the energy stored in a battery, extending its operational lifetime.
- VBAT Input range: 1.8 to 3.8 V.
- VDC Output range: 1.25 to 3.8 V in 50 mV (1.25–1.8 V) or 100 mV (1.8–3.8 V) steps.
- Supplies up to 100 mA.
- Internal voltage reference.
- Converter clock source selectable between the divided APB clock or a dedicated local oscillator.
- Automatically limits the peak inductor current if the load current rises beyond a safe limit.
- Automatically goes into bypass mode if the battery voltage cannot provide sufficient headroom.
- Sources current, but cannot sink current.

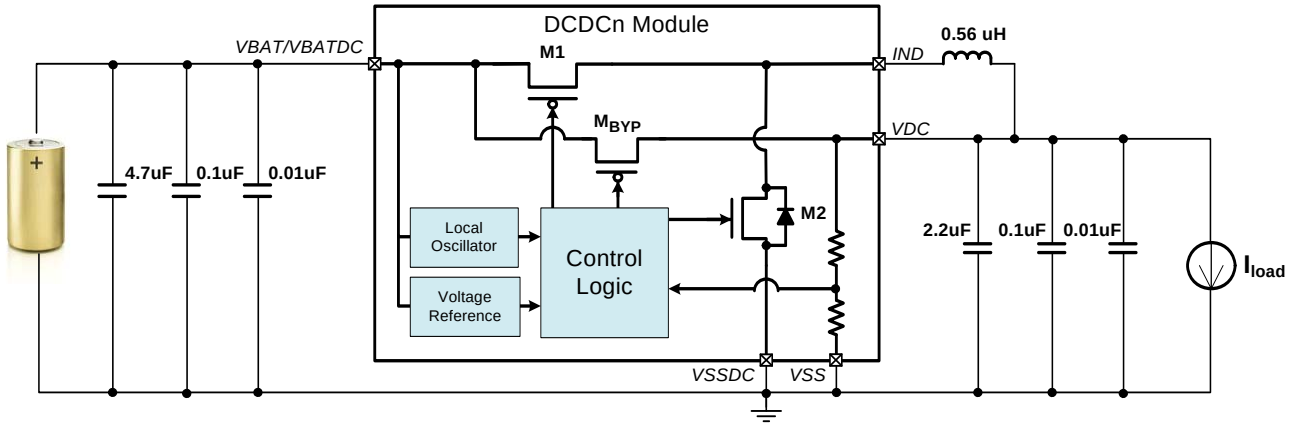


Figure 12.1. DCDC0 Block Diagram

12.2. Inductor Selection and Startup Behavior

The DCDC module requires clocks and bias voltages from several modules in the device. The APB clock enable bits DCDC0CEN, LCD0CEN and MISC0CEN must be set to 1 and the dc-dc bias in the LCD0 block (LCD0_CONFIG0.DCDCBIASEN) must be enabled. The dc-dc converter may then be enabled by setting the DCDCEN bit in the CONTROL register to logic 1.

When first enabled, the dc-dc converter will control the M1 and M2 switches to supply current into the output capacitor through the inductor until the VDC output voltage reaches the programmed level set by the OUTVSEL field in the CONTROL register.

The RDYHIGHF and RDYLOWF flags in the CONTROL register may be used to determine when the output voltage is near the programmed level. RDYHIGHF is a fixed limit, and will be set high when the output is above 105% of the programmed level. RDYLOWF will be set when the output voltage is above the level defined by the RDYLOWTH field in the CONFIG register.

The dc-dc converter is designed to provide up to 100 mA of output current.

12.2.1. Inductor Selection

In order to minimize power loss and maximize efficiency, a 0.56 µH inductor with a dc resistance of 500 mΩ or less is recommended. Example inductor part numbers are listed in Table 12.1.

Table 12.1. Example DC-DC Inductors

Manufacturer	Part Number	L (µH)	Max. DCR (mΩ)	Peak Current Rating (mA)
Taiyo Yuden	BRC1608TR56M	0.56	123.5	1,150
Panasonic	ELJ-FBR56MF	0.56	420	560
Murata	LQM2HPNR56ME0L	0.56	75	1,500

12.2.2. Peak Inductor Current

The peak transient current in the inductor is limited for safe operation. The peak inductor current is programmable using the ILIMIT field in register CONFIG. The peak inductor current, size of the output capacitor and the amount of dc load current present during startup will determine the length of time it takes to charge the output capacitor. In order to ensure reliable startup of the dc-dc converter, the following restrictions have been imposed:

- The maximum dc load current allowed during startup is given in the electrical specification tables in the device data sheet. If the dc-dc converter is powering external sensors or devices through the VDC pin, then the current supplied to these sensors or devices is counted towards this limit. The in-rush current into capacitors does not count towards this limit.
- The maximum total output capacitance is also given in the electrical specification tables. This value includes the required 2.2 µF ceramic output capacitor and any additional capacitance connected to the VDC pin.

The peak inductor current limit is programmable by software from 200 mA to 800 mA via the ILIMIT field in the CONFIG register. Limiting the peak inductor current can allow the dc-dc converter to start up using a high impedance power source (such as when a battery is near its end of life) or allow inductors with a low current rating to be utilized.

The peak inductor current is dependent on several factors including the dc load current and can be estimated using the following equation:

$$I_{PK} = \sqrt{\frac{2 \times I_{LOAD} \times (VBATDC - VDC)}{inductance \times frequency} \times \frac{VDC}{VBATDC}}$$

where inductance = 0.56 µH and frequency = 1.9 to 3.9 MHz.

12.2.3. DC-DC Startup Procedure (DC-DC Clock Source = Local Oscillator)

1. Enable the APB clock to the DCDC0, LDC0, and MISC0.
2. Enable the dc-dc bias enable in the LCD module (LCD0_CONFIG0.DCDCBIASEN = 1)
3. Clear the local oscillator disable bit to enable the local oscillator (OSCDIS = 0).
4. Clear the clock source select bit to set the dc-dc clock source to local oscillator (CLKSEL = 0).
5. Set the inductor peak current limit bits (ILIMIT) according to the inductor peak current rating.
6. Set the converter ready low threshold (RDYLOWTH) to desired threshold.
7. Set the power switch mode bits (PSMD) to desired value.
8. Set the output voltage select bits (OUTVSEL) to the desired output voltage.
9. Enable the dc-dc converter (DCDCEN = 1);
10. Poll the dc-dc converter ready low flag (RDYLOWF) until it reads 1.
11. Check that the dc-dc converter ready high flag (RDYHIGHF) reads 0.

12.2.4. DC-DC Startup Procedure (DC-DC Clock Source = APB Clock)

1. Enable the APB clock to the DCDC0, LDC0, and MISC0.
2. Enable the dc-dc bias enable in the LCD module (LCD0_CONFIG0.DCDCBIASEN = 1)
3. Set the local oscillator disable bit to disable the local oscillator (OSCDIS = 1).
4. Set the clock divider bits (CLKDIV) to an appropriate divider to ensure that the dc-dc clock is in the range of 1.9 MHz to 3.9 MHz. For example, if the APB clock = 49 MHz, set the CLKDIV = 4 to configure the dc-dc clock to APB clock / 16, or 3.06 MHz.
5. Set the clock source select bit to set the dc-dc clock source to the APB clock (CLKSEL = 1).
6. Set the inductor peak current limit bits (ILIMIT) according to the inductor peak current rating.
7. Set the converter ready low threshold (RDYLOWTH) to desired threshold.
8. Set the power switch mode bits (PSMD) to desired value.
9. Set the output voltage select bits (OUTVSEL) to the desired output voltage.
10. Enable the dc-dc converter (DCDCEN = 1);
11. Poll the dc-dc converter ready low flag (RDYLOWF) until it reads 1.
12. Check that the dc-dc converter ready high flag (RDYHIGHF) reads 0.

12.3. Synchronous/Asynchronous Modes

The dc-dc converter provides both synchronous and asynchronous switching modes, controlled by the ASYNCEN bit in the CONTROL register. In synchronous mode (ASYNCEN=0), the dc-dc converter will switch both internal MOSFETs, M1 and M2. In asynchronous mode (ASYNCEN=1), the M2 MOSFET is disabled and current conducts through the M2 MOSFET parasitic diode. Synchronous mode is more efficient for nominal to heavy output loads, and asynchronous mode is more efficient for very light loads. Synchronous mode serves to emulate a rectification diode with low voltage-drop. As such, the DC-DC converter operates in discontinuous conduction mode (DCM) in both asynchronous and synchronous switching modes.

Consult Table 12.2 for guidance on ASYNCEN settings for various load currents.

12.4. Pulse Skipping

The dc-dc converter allows the user to set the minimum pulse width such that if the duty cycle needs to decrease below a certain width in order to maintain regulation, an entire "clock pulse" will be skipped. Pulse skipping is controlled by the MINPWSEL field in the CONTROL register.

Pulse skipping can provide substantial power savings, particularly at low values of load current. The converter will continue to maintain the output voltage at its programmed value when pulse skipping is employed, though the output voltage ripple can be higher. Another consideration is that the dc-dc will operate with pulse-frequency modula-

tion rather than pulse-width modulation, which makes the switching frequency spectrum less predictable; this could be an issue if the dc-dc converter is used to power a radio.

Consult Table 12.2 for guidance on MIWPWSEL settings for various load currents.

12.5. Power Switch Size

The dc-dc converter's M1 and M2 switches consist of up to four MOSFETs in parallel, with the number of MOSFETs actively driven is determined by the Power Switch Mode (PSMD) field in the CONTROL register. When PSMD = 0, the switches use only one MOSFET, resulting in higher conduction loss (i.e., loss due to power dissippated across the switch ON resistance) but lower switching loss (i.e., loss due to the power required to drive the switch gates to change the state of the MOSFET). When PSMD = 3, the switches consist of four MOSFETs in parallel, resulting in lower conduction loss and higher switching loss.

Because conduction loss increases with output current, at high output load currents the conduction loss will dominate the dc-dc converter losses. For high output currents, the PSMD should be set to 2 or 3 to minimize the switch ON resistance and the conduction losses.

At low output load currents, the conduction loss is very low and the dc-dc converter losses should typically be dominated by the switching losses. For low output currents, the PSMD should be set to 0 or 1 to minimize the switching losses.

Consult Table 12.2 for guidance on PSMD settings for various load currents.

12.6. DC-DC Configuration Guidelines

Table 12.2 provides dc-dc configuration recommendations over the range of output loads. These guidelines provide a reasonable tradeoff between optimal efficiency and simplicity..

Table 12.2. Optimizing DC-DC Efficiency, VBAT=3.8V, DC-DC Clock = 2.6MHz

Output Load Current (mA)	PSMD	ASYNCEN	MINPWSEL
ILOAD $\geq$ 15	3	0	0
5 $\leq$ ILOAD < 15	0	0	0
ILOAD < 5	0	1	3

12.7. Optimizing Board Layout

The PCB layout does have an effect on the overall efficiency. The following guidelines are recommended to achieve the optimum layout:

- Place the input capacitor stack as close as possible to the VBAT/VBATDC pin. The smallest value capacitors in the stack should be placed closest to the VBAT/VBATDC pin.
- Place the output capacitor stack as close as possible to the VDC pin. The smallest value capacitors in the stack should be placed closest to the VDC pin.
- Minimize the trace length and trace impedance between the IND pin, the inductor, and the VDC pin.

12.8. DC-DC Converter Clocking Options

The dc-dc converter may be clocked from its internal oscillator, or from any system clock source, selectable by the CLKSEL bit in the CONTROL register. The dc-dc converter internal oscillator frequency is approximately 2.9 MHz. For a more accurate clock source, the APB clock, or a divided version of the APB clock may be used as the dc-dc clock source. When selecting the APB-derived clock, the CLKDIV field in the CONTROL register should be configured to supply a clock in the range of 1.9 MHz to 3.9 MHz.

To minimize interference in noise-sensitive applications, the DCDC block includes some additional clock controls. The clock routed to the dc-dc converter clock divider may be inverted by setting the CLKINVEN bit to logic 1, shifting the edge on which the DCDC operates. To minimize the effects of the DCDC switching on SARADC conversions, the ADCSYNCEN bit may be used. When enabled, this feature synchronizes the SARADC clock to the DCDC clock and causes the ADC to track during quiet times in the DCDC switching period. The polarity of the clock provided to the ADC can also be inverted using the ADCCLKINVEN bit.

12.9. Bypass Mode

The dc-dc converter has a bypass switch (MBYP), shown in Figure 12.1, which allows the output voltage (VDC) to be directly tied to the input supply (VBAT/VBATDC), bypassing the dc-dc converter. The bypass switch may be used independently from the dc-dc converter. For example, applications that need to power the VDC supply in the lowest power Sleep mode can turn on the bypass switch prior to turning off the dc-dc converter in order to avoid powering down the external circuitry connected to VDC.

There are two ways to close the bypass switch. Using the first method, Forced Bypass Mode, the BEN bit in the CONTROL register is set to a logic 1 forcing the bypass switch to close. Clearing the BEN bit to logic 0 will allow the switch to open if it is not being held closed using Automatic Bypass Mode.

The Automatic Bypass Mode, enabled by setting the ABEN bit to logic 1, closes the bypass switch when the difference between VBAT/VBATDC and the programmed output voltage is less than approximately 0.4 V. Once the difference exceeds approximately 0.5 V, the bypass switch is opened unless being held closed by Forced Bypass Mode. In most systems, Automatic Bypass Mode will be left enabled, and the Forced Bypass Mode may be used to close the switch as needed by the system.

12.10. Interrupts

The module interrupt enable (MIEN) bit in the CONTROL register is used to enable interrupts for the dc-dc module. The interrupt mode (INTMD) field in the CONFIG register determines the conditions which will generate system interrupts. Interrupts can be generated when the output voltage is too low, too high, in regulation, or out of regulation. Table 12.3 shows the mapping of INTMD settings to the RDYLOWF and RDYHIGHF states...

Table 12.3. Interrupt Generation Conditions

Interrupt Mode (INTMD)	Interrupt Generation Condition	Description
0x00	RDYLOWF=0	Output voltage is too low.
0x01	RDYLOWF=1	Output voltage is not too low.
0x10	RDYLOWF=0 OR RDYHIGHF=1	Output voltage is too high or too low.
0x11	RDYLOWF=1 AND RDYHIGHF=0	Output voltage is in regulation.

12.11. DCDC0 Registers

This section contains the detailed register descriptions for DCDC0 registers.

Register 12.1. DCDC0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DCDCEN	BEN	ABEN	ASYNCEN	PSMD		MINPWSEL		Reserved	MIEN	Reserved	OUTVSEL				
Type	RW	RW	RW	RW	RW		RW		R	RW	R	RW				
Reset	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ADCCLKINVEN	CLKINVEN	ADCSYNCEN	CLKDIV			CLKSEL	OSCDIS	Reserved				BGRDYF	DROPOUTF	RDYHIGHF	RDYLOWF
Type	RW	RW	RW	RW			RW	RW	R				R	R	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DCDC0_CONTROL = 0x4004_E000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 12.4. DCDC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	DCDCEN	**DC-DC Converter Enable.** 0: Disable the DC-DC converter. 1: Enable the DC-DC converter.
30	BEN	**Bypass Enable.** Setting this bit to 1 forces the MBYP switch on, connecting VBATDC to VDC. 0: Disable the MBYP bypass switch. 1: Enable the MBYP bypass switch.
29	ABEN	**Automatic Bypass Enable.** If this bit is set to 1, the MBYP switch automatically turns on if the converter is in dropout (DROPOUTF = 1). 0: Disable automatic bypass. 1: Enable automatic bypass.

Table 12.4. DCDC0_CONTROL Register Bit Descriptions

Bit	Name	Function
28	ASYNCEN	**Asynchronous Mode Enable.** 0: Enable DC-DC synchronous mode. 1: Enable DC-DC asynchronous mode. This mode is more efficient for very light output loads.
27:26	PSMD	**Power Switch Mode.** This field selects mode of the M1 and M2 power switches in the converter. Using lower modes results in higher efficiency at low supply currents. 00: Mode 0. Set the M1 and M2 power switches to each use one MOSFET only. 01: Mode 1. Set the M1 and M2 power switches to each use 2 MOSFETS in parallel. 10: Mode 2. Set the M1 and M2 power switches to each use 3 MOSFETS in parallel. 11: Mode 3. Set the M1 and M2 power switches to each use 4 MOSFETS in parallel.
25:24	MINPWSEL	**Minimum Pulse Width Select.** 00: Disable pulse skipping. 01: Set the minimum pulse width to 10 ns. 10: Set the minimum pulse width to 20 ns. 11: Set the minimum pulse width to 40 ns.
23	Reserved	Must write reset value.
22	MIEN	**Module Interrupt Enable.** 0: Disable DC-DC module interrupts. 1: Enable DC-DC module interrupts.
21	Reserved	Must write reset value.
20:16	OUTVSEL	**Output Voltage Select.** This field determines the output voltage of the DC-DC converter. A value of 0 corresponds to 1.25 V, and a value of 31 corresponds to 3.8 V. When the value of OUTVSEL is less than 11 (0x0B), the output voltage step size is 50 mV. Otherwise, the step size is 100 mV.
15	ADCCLKINVEN	**ADC Clock Inversion Enable.** 0: Do not invert the ADC clock derived from the DC-DC switching frequency. 1: Invert the ADC clock derived from the DC-DC switching frequency.
14	CLKINVEN	**Clock Inversion Enable.** When using the APB clock as the converter clock source (CLKSEL = 1), setting this bit inverts the system clock input.
13	ADCSYNCEN	**ADC Synchronization Enable.** When this bit is set to 1, the ADC tracks during the longest quiet time of the DC-DC converter switching cycle, and the ADC clock is also synchronized to the DC-DC converter switching cycle. The CLKDIV field in the ADCn module must be cleared to 0 when synchronization is enabled. 0: Do not synchronize the ADC to the DC-DC converter. 1: Synchronize the ADC to the DC-DC converter.

Table 12.4. DCDC0_CONTROL Register Bit Descriptions

Bit	Name	Function
12:10	CLKDIV	**Clock Divider.** When CLKSEL is set to 1, setting this field divides the APB clock input to the DC-DC converter. The converter switching frequency is limited to 1.9 MHz to 3.9 MHz. If the APB is running at a higher frequency, it must be appropriately divided so that the converter switching frequency is within the range. 000: Use the APB clock divided by 1 as the converter switching frequency. 001: Use the APB clock divided by 2 as the converter switching frequency. 010: Use the APB clock divided by 4 as the converter switching frequency. 011: Use the APB clock divided by 8 as the converter switching frequency. 100: Use the APB clock divided by 16 as the converter switching frequency. 101-111: Reserved.
9	CLKSEL	**Clock Source Select.** 0: Select the local DC-DC oscillator as the clock source. 1: Select the APB clock as the clock source.
8	OSCDIS	**Oscillator Disable.** 0: Enable the DC-DC local oscillator. 1: Disable the DC-DC local oscillator.
7:4	Reserved	Must write reset value.
3	BGRDYF	**Bandgap Ready Flag.** 0: The bandgap voltage is not above the threshold. 1: The bandgap voltage is above the threshold.
2	DROPOUTF	**DC-DC Converter Dropout Flag.** 0: The input voltage (VBATDC) is more than 0.4 V above the output voltage (VDC). The DC-DC converter is not in dropout. 1: The input voltage (VBATDC) is less than 0.4 V above the output voltage (VDC). The DC-DC converter is in dropout, and firmware should enable the bypass switch (BEN=1).
1	RDYHIGHF	**DC-DC Converter Ready High Flag.** 0: The output voltage (VDC) has not exceeded 105% of the programmed output value. 1: The output voltage (VDC) has exceeded 105% of the programmed output value.
0	RDYLOWF	**DC-DC Converter Ready Low Flag.** 0: The output voltage (VDC) is below the threshold set in the RDYLOWTH threshold field (RDYLOWTH). 1: The output voltage (VDC) is above the threshold set in the RDYLOWTH threshold field (RDYLOWTH).

Register 12.2. DCDC0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved										RDYLOWTH		Reserved		INTMD	
Type	R										RW		R		RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									ILIMIT			Reserved			
Type	R		RW						R	RW			R		RW	
Reset	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0	0

Register ALL Access Address

DCDC0_CONFIG = 0x4004_E010

Table 12.5. DCDC0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:22	Reserved	Must write reset value.
21:20	RDYLOWTH	**Converter Ready Low Threshold.** 00: Hardware sets the RDYLOWF flag if the regulated output voltage is greater than 95% of the programmed output voltage. 01: Hardware sets the RDYLOWF flag if the regulated output voltage is greater than 90% of the programmed output voltage. 10: Hardware sets the RDYLOWF flag if the regulated output voltage is greater than 85% of the programmed output voltage. 11: Hardware sets the RDYLOWF flag if the regulated output voltage is greater than 80% of the programmed output voltage.
19:18	Reserved	Must write reset value.
17:16	INTMD	**Interrupt Mode.** This field determines the condition under which a DC-DC converter interrupt occurs, if enabled (MIEN = 1). 00: Generate an interrupt when the regulated converter output voltage is too low, according to the RDYLOWF flag. 01: Generate an interrupt when the regulated converter output voltage is not too low according to the RDYLOWF flag. 10: Generate an interrupt when the output voltage is out of regulation. The converter output can be either too high or too low, according to the RDYLOWF and RDYHIGHF flags. 11: Generate an interrupt when the output voltage is in regulation.
15:7	Reserved	Must write reset value.

Table 12.5. DCDC0_CONFIG Register Bit Descriptions

Bit	Name	Function
6:4	ILIMIT	**Inductor Peak Current Limit.** 000: Reserved. 001: Limit the peak inductor current to 200 mA. 010: Limit the peak inductor current to 300 mA. 011: Limit the peak inductor current to 400 mA. 100: Limit the peak inductor current to 500 mA. 101: Limit the peak inductor current to 600 mA. 110: Limit the peak inductor current to 700 mA. 111: Limit the peak inductor current to 800 mA.
3:0	Reserved	Must write reset value.

12.12. DCDC0 Register Memory Map

Table 12.6. DCDC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DCDC0_CONTROL	0x4004_E000	ALL | SET | CLR	DCDCEN	BEN	ABEN	ASYNCEN	PSMD		MINPWSEL		Reserved	MIEN	Reserved		OUTVSEL				ADCCLKINVEN	CLKINVEN	ADCSYNCEN	CLKDIV			CLKSEL	OSCDIS	Reserved				BGRDYF	DROPOUTF	RDYHIGHF	RDYLOWF
DCDC0_CONFIG	0x4004_E010	ALL	Reserved										RDYLOWTH		Reserved		INTMD		Reserved							ILIMIT				Reserved				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

13. Device Identification (DEVICEID0)

This section describes the Device Identification (DEVICEID) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

13.1. Device ID Features

The Device ID module includes the following features:

- Unique 4-word Device ID.

The unique 4-word device ID contains 124 bits of UUID data and the device revision number.

13.2. DEVICEID0 Registers

This section contains the detailed register descriptions for DEVICEID0 registers.

Register 13.1. DEVICEID0_DEVICEID0: Device ID Word 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEVICEID0[27:12]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DEVICEID0[11:0]												REVID			
Type	RW												RW			
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

DEVICEID0_DEVICEID0 = 0x4004_90C0

Table 13.1. DEVICEID0_DEVICEID0 Register Bit Descriptions

Bit	Name	Function
31:4	DEVICEID0	**Device ID 0.**
3:0	REVID	**Revision ID.** This field provides the revision information for the device. 0000: Revision A. 0001: Revision B. 0010-1111: Reserved.

Register 13.2. DEVICEID0_DEVICEID1: Device ID Word 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEVICEID1[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DEVICEID1[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Register ALL Access Address DEVICEID0_DEVICEID1 = 0x4004_90D0																

Table 13.2. DEVICEID0_DEVICEID1 Register Bit Descriptions

Bit	Name	Function
31:0	DEVICEID1	**Device ID 1.**

Register 13.3. DEVICEID0_DEVICEID2: Device ID Word 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEVICEID2[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DEVICEID2[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Register ALL Access Address DEVICEID0_DEVICEID2 = 0x4004_90E0																

Table 13.3. DEVICEID0_DEVICEID2 Register Bit Descriptions

Bit	Name	Function
31:0	DEVICEID2	**Device ID 2.**

Register 13.4. DEVICEID0_DEVICEID3: Device ID Word 3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEVICEID3[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DEVICEID3[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Register ALL Access Address DEVICEID0_DEVICEID3 = 0x4004_90F0																

Table 13.4. DEVICEID0_DEVICEID3 Register Bit Descriptions

Bit	Name	Function
31:0	DEVICEID3	**Device ID 3.**

13.3. DEVICEID0 Register Memory Map

Table 13.5. DEVICEID0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DEVICEID0_DEVICEID0	0x4004_90C0	ALL	DEVICEID0																												REVID			
DEVICEID0_DEVICEID1	0x4004_90D0	ALL	DEVICEID1																															

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 13.5. DEVICEID0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DEVICEID0_DEVICEID2	0x4004_90E0	ALL	DEVICEID2																															
DEVICEID0_DEVICEID3	0x4004_90F0	ALL	DEVICEID3																															

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

14. Advanced Capture Counter (ACCTR0)

This section describes the Advanced Capture Counter (ACCTR) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the ACCTR block, which is used by all device families covered in this document.

14.1. ACCTR Features

The SiM3L1xx devices contain a low-power Advanced Capture Counter module that runs from the RTC0 timer clock and can be used with digital inputs, switch topology circuits (reed switches), or with LC resonant circuits. For switch topology circuits, the module charges one or two external lines by pulsing internal pull-up resistors and detecting whether the reed switch is open or closed. For LC resonant circuits, the inputs are periodically energized to produce a dampened sine wave and configurable discriminator circuits detect the resulting waveform decay.

The ACCTR module has the following general features:

- Single or differential inputs supporting single, dual, and quadrature modes of operation.
- Variety of interrupt and PM8 wake up sources.
- Provides feedback of the direction history, current and previous states, and condition flags.

The ACCTR module has the following features for switch circuit topologies:

- Ultra low power input comparators.
- Supports a wide range of pull-up resistor values with a self-calibration engine.
- Asymmetrical integrators for low-pass filtering and switch debounce.
- Two 24-bit counters and two 24-bit digital threshold comparators.
- Supports switch flutter detection.

For LC resonant circuit topologies, the ACCTR module includes:

- Separate minimum and maximum count registers and polarity, pulse, and toggle controls.
- Zone-based programmable timing.
- Two input comparators with support for a positive side input bias at VIO divided by 2.
- Supports a configurable excitation pulse width based on an internal 40 MHz oscillator and timer or an external digital stop signal.
- Two 8-bit peak counters that saturate at full scale for detecting the number of LC resonant peaks.
- Two discriminators with programmable thresholds.
- Supports a sample and hold mode for Wheatstone bridges.

14.2. Overview

The SiM3L1xx family of microcontrollers contains a low-power Advanced Capture Counter (ACCTR) module designed to count pulses from many different types of sources including digital inputs, switch topology circuits (reed switches), LC resonant circuits and Wheatstone bridges. For switch topology circuits, it charges 1 or 2 external lines by pulsing different size pull up resistors and then detects the reed switch's open or close state. It also supports external LC resonant circuits which are periodically energized to produce a dampened sine wave that can be used to count the number of peaks over a programmable threshold. A discriminator circuit then decides if a rotating wheel is in the dampened or non-dampened region based on the number of counted peaks being either above or below programmable digital thresholds. The ACCTR module also includes asymmetrical integrators for low pass filtering and switch debounce, single, dual, and quadrature modes of operation, flutter detection, two 24-bit counters, two 24-bit threshold comparators, and a variety of interrupt and sleep wake up capabilities. This combination of features provides water, gas, and heat metering system designers with an optimal tool for saving power while collecting meter usage data.

The ACCTR can operate in power mode 8 (PM8) to enable ultra-low power metering systems. The MCU does not have to wake up on every edge or transition and can remain in PM8 while the ACCTR counts pulses for an extended period of time. The ACCTR includes two 24-bit counters. These counters can count up to 16,777,215 (2^{24}-1) transitions in sleep mode before overflowing. The ACCTR can wake up the MCU when one of the counters overflows. The ACCTR also has two 24-bit digital comparators. The digital comparators have the ability to wake up the MCU when either of the counters reaches a predetermined threshold.

The ACCTR uses the RTC timer clock for sampling, de-bouncing, managing the low-power pull-up resistors, and timing of stimulus pulses. The RTC0TCLK signal must be enabled when counting pulses. If desired, the RTC alarms can wake up the MCU periodically to read the pulse counters, instead of using the digital comparators. For example, the RTC can wake up the MCU every five minutes. The MCU can then read the count values and transmit the information using the UART or a wireless transceiver before returning to sleep.

14.3. External Pin Connections

The external pin connections for the ACCTR module include up to four analog I/O and up to eight digital I/O. Not all pins are used in every mode or with every external circuit configuration. Any pins used by the ACCTR module should be configured to the appropriate mode (analog or digital) in the PBCFG module and skipped by the crossbar.

Table 14.1. ACCTR0 External Pin Connections

ACCTR0 Signal Name	Type	Function	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
ACCTR0_IN0	Analog In	Switch Input 1 LC Comp0-	PB0.5	PB0.4	PB0.4
ACCTR0_IN1	Analog In	Switch Input 0 LC Comp1-	PB0.6	PB0.5	PB0.5
ACCTR0_STOP0	Digital In	LC Channel 0 Pulse Stop	PB0.5	PB0.4	Reserved
ACCTR0_STOP1	Digital In	LC Channel 1 Pulse Stop	PB0.6	PB0.5	Reserved
ACCTR0_LCIN0	Analog In	LC Comp0+	PB0.7	PB0.6	Reserved
ACCTR0_LCIN1	Analog In	LC Comp1+	PB0.8	PB0.7	Reserved
ACCTR0_LCPUL0	Digital Out	LC Channel 0 Pulse Out	PB0.9	PB0.8	Reserved
ACCTR0_LCPUL1	Digital Out	LC Channel 1 Pulse Out	PB0.10	PB0.9	Reserved
ACCTR0_LCBIAS0	Digital Out	LC Channel 0 Bias Out	PB1.0	PB1.0	Reserved
ACCTR0_LCBIAS1	Digital Out	LC Channel 1 Bias Out	PB1.1	PB1.1	Reserved
ACCTR0_DEBUG0	Digital Out	Debug Output 0	PB1.4	PB1.4	Reserved
ACCTR0_DEBUG1	Digital Out	Debug Output 1	PB1.5	PB1.5	Reserved

14.4. Analog Front End

The analog front end of the ACCTR module supports a wide variety of external circuits. The ACCTR module supports two major analog input modes: switch topology mode, and LC resonant mode. Switch topology mode is typically used for simpler circuits such as reed switches. The LC resonant mode is more complex, and can be used with LC resonant circuits, capacitive circuits, Wheatstone bridges, continuous variable-frequency pulse trains, etc. The main focus in this documentation is on reed switch usage and LC resonant circuit usage, though other external circuits may be mentioned. The TOPMD bit in the CONFIG register selects between switch topology mode and LC resonant mode.

14.4.1. Switch Topology Mode

The Advanced Capture Counter works with both Form-A and Form-C reed switches. A Form-A switch is a Normally-Open Single-Pole Single-Throw (NO SPST) switch. A Form-C reed switch is a Single-Pole Double-Throw (SPDT) switch. Figure 14.1 illustrates some of the common reed switch configurations for a single-channel meter.

The Form-A switch requires a pull-up resistor. The energy used by the pull-up resistor may be a substantial portion of the energy budget. To minimize energy usage, the Advanced Capture Counter has a programmable pull-up resistance and an automatic calibration engine. The calibration engine can automatically determine the smallest usable pull-up strength setting. A Form-C switch does not require a pull-up resistor and will provide a lower power solution. However, the Form-C switches are more expensive and require an additional wire for VBAT.

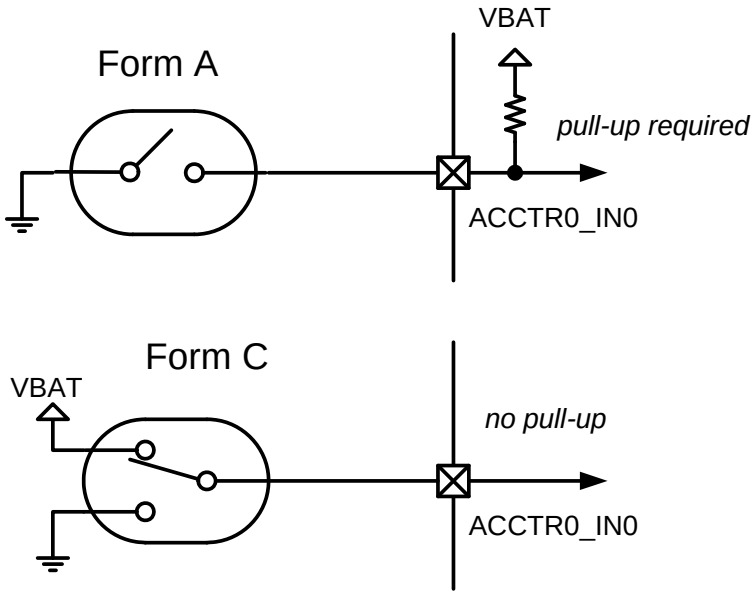


Figure 14.1. Reed Switch Configurations

14.4.1.1. Programmable Pull-Up Resistors

The Advanced Capture Counter features low-power pull-up resistors with a programmable resistance and duty-cycle. The average pull-up current will depend on the selected resistor, sample rate, and pull-up duty-cycle. The PUVAL field in the CONTROL register adjusts the value and duty cycle of the internal pull-ups.

Table 14.2 through Table 14.5 give the average current for all combinations of PUVAL at certain sampling rates.

Table 14.2. Average Pull-Up Current (Sample Rate = 250 µs)

	PUVAL[4:2]								Duty Cycle
PUVAL[1:0]	000	001	010	011	100	101	110	111	
00	disabled	250 nA	1.0 µA	4.0 µA	16 µA	64 µA	250 µA	1000 µA	25%
01	disabled	375 nA	1.5 µA	6.0 µA	24 µA	96 µA	375 µA	1500 µA	37.5%
10	disabled	500 nA	2.0 µA	8.0 µA	32 µA	128 µA	500 µA	2000 µA	50%
11	disabled	750 nA	3.0 µA	12.0 µA	48 µA	192 µA	750 µA	3000 µA	75%

Table 14.3. Average Pull-Up Current (Sample Rate = 500 µs)

	PUVAL[4:2]								Duty Cycle
PUVAL[1:0]	000	001	010	011	100	101	110	111	
00	disabled	125 nA	0.50 µA	2.0 µA	8 µA	32 µA	125 µA	500 µA	12.5%
01	disabled	188 nA	0.75 µA	3.0 µA	12 µA	48 µA	188 µA	750 µA	18.8%
10	disabled	250 nA	1.0 µA	4.0 µA	16 µA	64 µA	250 µA	1000 µA	25%
11	disabled	375 nA	1.5 µA	6.0 µA	24 µA	96 µA	375 µA	1500 µA	37.5%

Table 14.4. Average Pull-Up Current (Sample Rate = 1 ms)

	PUVAL[4:2]								Duty Cycle
PUVAL[1:0]	000	001	010	011	100	101	110	111	
00	disabled	63 nA	250 nA	1.0 µA	4 µA	16 µA	63 µA	250 µA	6.3%
01	disabled	94 nA	375 nA	1.5 µA	6 µA	24 µA	94 µA	375 µA	9.4%
10	disabled	125 nA	500 nA	2.0 µA	8 µA	32 µA	125 µA	500 µA	12.5%
11	disabled	188 nA	750 nA	3.0 µA	12 µA	48 µA	188 µA	750 µA	18.8%

Table 14.5. Average Pull-Up Current (Sample Rate = 2 ms)

	PUVAL[4:2]								Duty Cycle
PUVAL[1:0]	000	001	010	011	100	101	110	111	
00	disabled	31 nA	125 nA	0.50 µA	2.0 µA	8 µA	31 µA	125 µA	3.1%
01	disabled	47 nA	188 nA	0.75 µA	3.0 µA	12 µA	47 µA	188 µA	4.7%
10	disabled	63 nA	250 nA	1.0 µA	4.0 µA	16 µA	63 µA	250 µA	6.3%
11	disabled	94 nA	375 nA	1.5 µA	6.0 µA	24 µA	94 µA	375 µA	9.4%

14.4.1.2. Automatic Pull-Up Resistor Calibration

The Advanced Capture Counter includes an automatic calibration engine which can automatically determine the minimum pull-up current for a particular application. The automatic calibration is especially useful when the load capacitance of field wiring varies from one installation to another.

The automatic calibration uses one of the Advanced Capture Counter inputs (ACCTR0_IN0 or ACCTR0_IN1) for calibration. The CALSEL bit in the CONTROL SFR selects either ACCTR0_IN0 or ACCTR0_IN1 for calibration, and the CALPUMD and CALMD fields control how calibration will be performed. The reed switch on the selected input should be in the open state to allow the signal to charge during calibration. The calibration engine can calibrate the pull-ups with the meter connected normally, provided that the reed switch is open during calibration. During calibration, the integrators will ignore the input comparators, and the counters will not be incremented. Using a 250 µs sample rate and a 32 kHz RTC0TCLK, the calibration time will be 21 ms (28 tests @ 750 µs each) or shorter depending on the pull up strength selected. The calibration will fail if the reed switch remains closed during this entire period. If the reed switch is both opened and closed during the calibration period, the value written into PUVAL may be larger than what is actually required. The transition flag (TRANSI in STATUS) can detect when the reed switch opens, and most systems with a wheel rotation of 10 Hz or slower should have sufficient high time for the calibration to complete before the next closing of the reed switch. Slowing the sample rate will also increase the calibration time. The same drive strength will used for both ACCTR0_IN0 and ACCTR0_IN1.

14.4.2. LC Resonant Mode

A typical connection diagram for an LC resonant circuit is shown in Figure 14.2. In this diagram the comparator is being used in a single-ended manner. It is possible to use the IN0 and IN1 inputs as the negative input to the comparator for differential topologies, such as magnetoresistive Wheatstone bridges.

In this circuit, the external resonant circuit is pulsed with an LC pulse signal (LCPUL) signal, and counters detect the number of peaks from a dampened sine wave at the LCIN input. A discriminator circuit compares the number of counts against a digital threshold to decide if the circuit is in the dampened or undampened region of a rotating wheel.

Note that the ESD protection in the input pads will clip voltages above 5.25 V and below –0.3 V. The LC comparator input can be externally ac coupled to the resonant circuit, and pre-biased at VIO/2 to limit clipping.

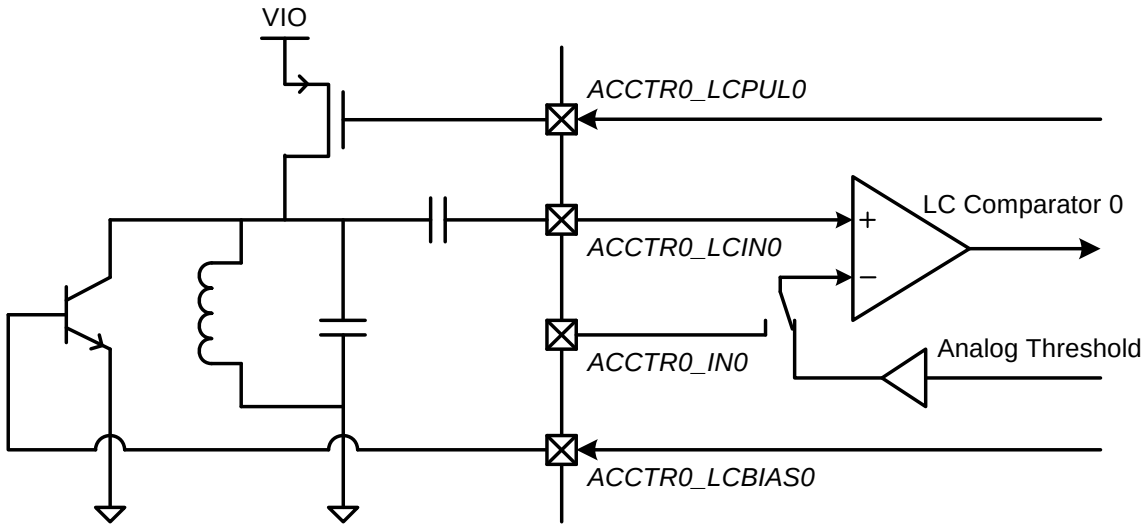


Figure 14.2. LC Resonant Configuration

14.5. Analog Comparator Functions

Four analog comparators are included in the ACCTR module, which should not be confused with the digital comparators in the ACCTR module nor with any other dedicated comparator peripherals on the device. Two of these (PC comparator 0 and PC comparator 1) are used in switch topology mode, while the other two (LC comparator 0 and LC comparator 1) are used in LC resonant mode. The difference in the two circuits lies primarily in their intended function. The switch topology comparators are designed with slower, high voltage signals in mind. They are much like a digital input with programmable VIH and VIL levels. The LC comparators are higher-precision circuits designed to quickly detect low-amplitude peaks of an oscillating signal. They can be operated as single-ended with a programmable internal threshold level, or differentially with two inputs.

14.5.1. Switch Topology Mode Comparators (PC comparator 0 and PC comparator 1)

When the ACCTR is configured in switch topology mode, the positive input of the PC comparators connect to the ACCTR0_INn pins. Channel 0 connects to ACCTR0_IN1, and channel 1 connects to ACCTR0_IN0. The input high and input low threshold levels are each configurable to four different levels via the CMPHTH and CMPLTH fields in the CONTROL register. Note that these two fields set the high and low thresholds for both of the PC comparators. The output of the comparator feeds directly into the integrator circuit in this mode. The output can also be routed to the ACCTR's debug output pins, or read from the CMP0OUT and CMP1OUT bits in the STATUS register.

14.5.2. LC Resonant Mode Comparators (LC comparator 0 and LC comparator 1)

When the ACCTR is configured in LC resonant mode, the positive input of the LC comparator connects to the corresponding ACCTR0_LCIN0 or ACCTR0_LCIN1 pin. If the LC resonant circuitry is configured for differential inputs (see Table 14.6), the negative input of the comparator will connect to the corresponding ACCTR0_IN0 or ACCTR0_IN1 pin. When configured for single-ended inputs, the negative terminal is connected to an internal adjustable threshold circuit. The threshold for the comparator is set by the CMPnTHR, CMPnCTH and CMPnFTH fields in the LCCONFIG register. By default, the LC comparators are only turned on when they are needed by the circuit, to save power. It is possible to force the comparators to an always on state using the FCMPnEN bits.

Additionally, the comparators support programmable hysterisis and speed. These features are controlled by the CMPHHYS, CMPLHYS and CMPMD fields in LCCONFIG. The hysterisis and speed settings are common to both comparators.

The output of the LC comparator feeds into the discriminator circuit in this mode. The output can also be routed to the ACCTR's debug output pins, or read from the CMP0OUT and CMP1OUT bits in the STATUS register.

14.6. LC Counting/Conditioning

In LC resonant mode the ACCTR can generate pulsed stimulus to excite external circuitry, and measure the resulting response characteristics. For an LC resonant circuit this takes the form of a dampened sinusoid. The comparator front end circuit detects the peaks of the sinusoid which occur over a short period of time, and a peak counter is used to accumulate the pulses. A discriminator circuit is used to compare the number of counts against a programmable threshold to determine whether the sinusoid is dampened or undampened.

14.6.1. LC Peak Counters

The LC peak counters count pulse outputs from the LC comparators. The LCCOUNT0 and 1 fields in the LCCOUNT register will contain the most recent LC counter value for each of the two channels.

14.6.2. LC Oscillator Calibration

The LC Oscillator is used to time several operations of the circuit in LC resonant mode. This oscillator nominally runs at 40 MHz and is only turned on to generate pulses as needed, to save power. Calibration of this oscillator can be performed in order to determine the actual oscillator frequency and adjust the timing parameters for the rest of the LC circuitry. An oscillator calibration does not change the frequency of the oscillator itself. Instead, the frequency is measured using the RTC0TCLK as a frequency reference. Setting the CLKCAL bit in the LCCLKCONTROL register to 1 initiates a calibration sequence. This bit will remain 1 during calibration, and be cleared to 0 when the calibration completes. During that time a special counter counts the number of LC Oscillator clock cycles which occur during one RTC0TCLK period. The result is presented in the CLKCYCLES field of LCCLKCONTROL.

14.6.3. Discriminators

The purpose of the discriminator is to make a distinction between how many counts of the LC peak counter constitute a logic 1 and logic 0. The discriminator consists of a digital threshold level with programmable hysterisis, and logic to enable automatic tracking and adjustment of the discriminator threshold. The current discriminator values for each channel are stored in the CD0 and CD1 field of the LCCOUNT register. The discriminator is enabled when bit 1 of the LCMD field is cleared to 0 (see Table 14.6). To operate without the discriminator (in sample-and-hold mode) bit 1 of the LCMD field can be set to 1.

14.6.3.1. Discriminator Digital Hysterisis

At the basic level, the discriminator can be set to a fixed value, acting as a threshold detector for 1s and 0's. Any value of the LC peak counter that is greater than or equal to the discriminator will result in a 0-to-1 transition at the discriminator output. Up to 3 counts of negative digital hysterisis can be applied to the high-to-low transition for each discriminator. The hysterisis is set by the LCD0HYS and LCD1HYS values in the LCMODE register.

14.6.3.2. Discriminator Tracking

The LCMODE register also contains two bits to enable automatic centering and signal tracking of the discriminator values. The ACDEN bit enables the automatic centering function. When centering is enabled, the discriminator value will be automatically updated to be centered between the MIN and MAX fields in the LCLIMITS register.

The MIN and MAX fields are updated after each count cycle has completed. By default, any value larger than the current MAX will replace MAX, and any value smaller than the current MIN will replace MIN. This behavior can be changed by enabling Automatic Tracking Mode using the ATRKEN bit. When ATRKEN is enabled, it forces the MIN and MAX values to remain the same distance apart, and only allows a change of one count per cycle. For example, if the current cycle completes and the count result is 5 codes higher than the current MAX value, MAX and MIN will both be incremented by 1 count. If the new count value is less than the current MIN value, both MIN and MAX are decremented by 1 count.

14.6.4. LC Pulse Stimulus

In LC resonant mode, the ACCTR has the ability to provide timed pulse stimuli to the external circuitry. A sampling event is divided into five distinct timing zones. The timing of the zones, the polarity of the pulses, and the region where comparisons are made are all programmable in firmware to achieve reliable, low-power operation for a variety of different circuit configurations.

14.6.4.1. Zones and TIming

The LC resonant stimulus circuitry divides the sampling event into a preconditioning zone (P) and four timing zones (A, B, C, D). Timing of each zone can be configured between 1 and 8 RTC0TCLK cycles using the ZONE fields in the TIMING register. The ACCTR0_LCPUL and ACCTR0_LCBIAS pins can be pulsed during selected zones to condition and excite external circuitry. The LC peak counters can then be activated during a selected zone to capture the resulting waveform. Figure 14.3 illustrates the timing zones with some example stimulus.

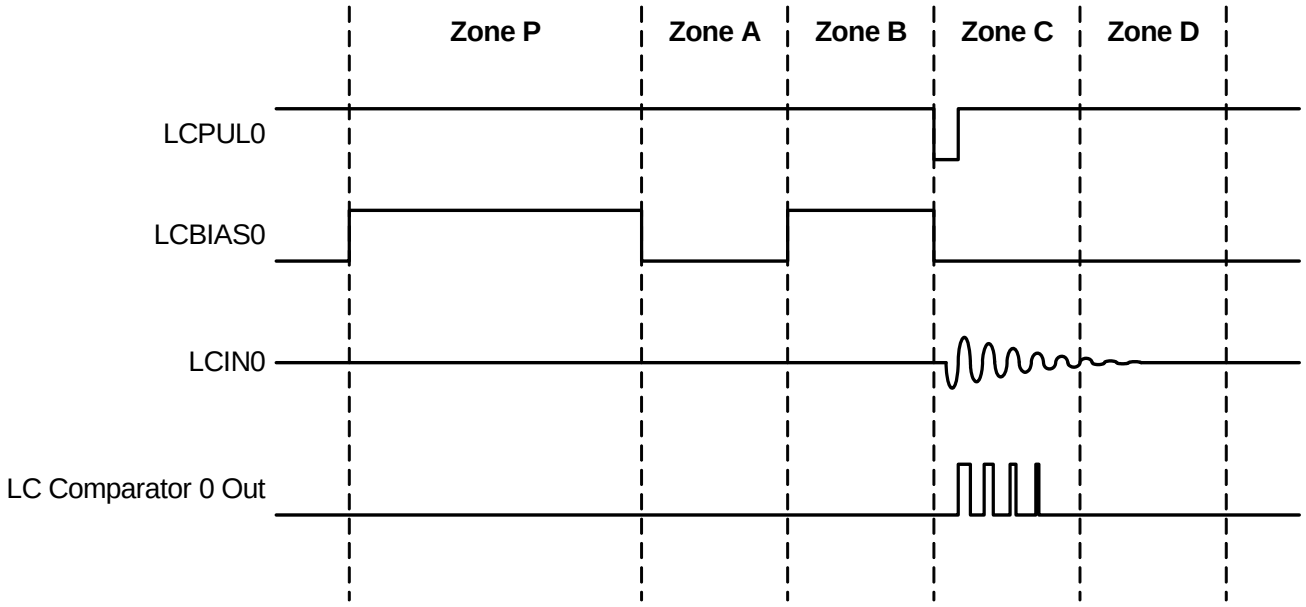


Figure 14.3. LC Resonant Timing

In this example, the bias signal LCBIAS0 is configured to pulse high during zones P and B. The LCPUL0 stimulus pulse is configured to pulse low during zone C and reset on an internal timer. The counter and comparator are enabled during zone C to capture the resulting waveform.

The bias pulsing options are straightforward. Bias pulses are generated at the LCBIAS pins with the polarity selected by the B0POL and B1POL bits. The user also has the option to generate bias on the external signals, internal signals, or both using the BMD field. Bias pulses are configured to occur in zones P, A, B or C using the corresponding bias zone enable bits in LCMODE. By default, bias pulses last the entire duration of the selected field. The bias pulses can be delayed from the beginning and end of the zone by 1/2 RTC0TCLK cycle each using the B0OEN and B1OEN bits in the TIMING register.

The pulse configuration options for the LCPUL pins are more extensive, and also more directly related to the operation of the rest of the circuit. A variety of different pulsing and measurement options are controlled by the LCMD field in the LCMODE register. These options are summarized in Table 14.6.

Table 14.6. LC Mode Options

LCMD Setting	Pulse Mode	Counter Behavior	Comparator Mode
0000	Full Zone	Discriminator	Single-Ended
0001	Full Zone	Discriminator	Differential
0010	Full Zone	Sample / Hold	Single-Ended
0011	Full Zone	Sample / Hold	Differential
0100	Stop on LC Timer	Discriminator	Single-Ended
0101	Stop on LC Timer	Discriminator	Differential
0110	Stop on LC Timer	Sample / Hold	Single-Ended
0111	Stop on LC Timer	Sample / Hold	Differential
1000	Stop on External Pin Rising Edge	Discriminator	Single-Ended
1001	Stop on External Pin Falling Edge	Discriminator	Single-Ended
1010	Stop on External Pin Rising Edge	Sample / Hold	Single-Ended
1011	Stop on External Pin Falling Edge	Sample / Hold	Single-Ended
1100	No Pulse Generated	Discriminator	Single-Ended
1101	No Pulse Generated	Discriminator	Differential
1110	No Pulse Generated	Sample / Hold	Single-Ended
1111	No Pulse Generated	Sample / Hold	Differential

The zone in which LCPUL pin pulses occur is selected between zones A, C, or A and C using the P0ZONE and P1ZONE fields, and the pulse type is selected between low, high and toggle operation using the PMD field.

LCMD[3:2] configures the type of pulse that will be generated on the LCPUL pin. Except when set to “No Pulse Generated”, the LCPUL pulse will begin at the start of the specified zone(s). When set to “Full Zone”, the LCPUL pulse ends at the end of the zone. When set to “Stop on LC Timer”, the pulse will last for the number of 40 MHz LC Oscillator cycles specified in the RELOAD field of LCCLKCONTROL. If set to “Stop on External Pin”, the pulse will last until the external STOP input meets the rising/falling edge condition.

The CnZONE fields in the LCTIMING register specify which zone the counter will operate in (A, B, C or D).

14.7. Counting Modes

The Advanced Capture Counter supports three different counting modes: single counter mode, dual counter mode, and quadrature counter mode. Figure 14.4 illustrates the three counter modes.

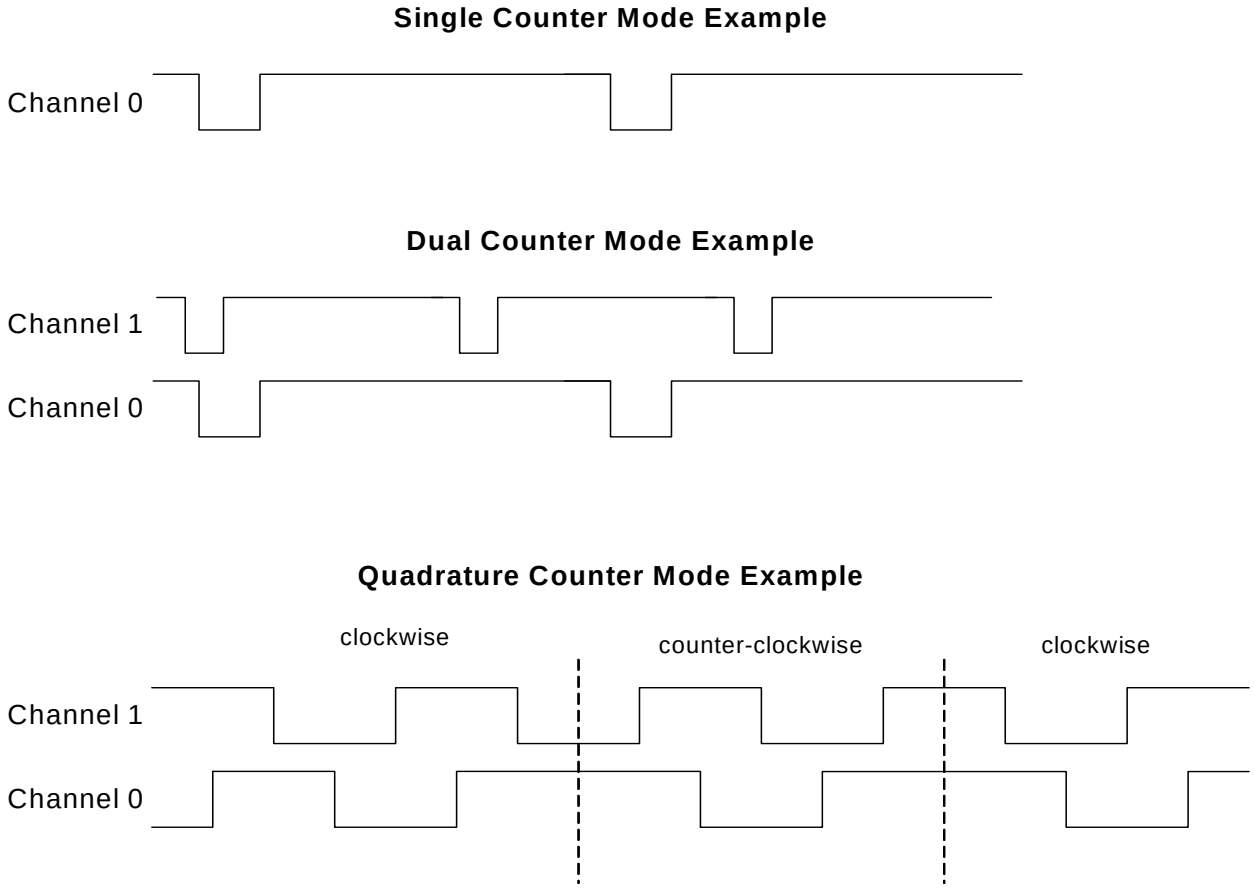


Figure 14.4. Counter Mode Examples

The single counter mode counts pulses from a single input channel. This mode uses only counter 0 and digital comparator 0 (counter 1 and digital comparator 1 are not used.) The single counter mode supports only one meter-encoder with a single-channel output. A single-channel encoder is an effective solution when the metered fluid flows only in one direction. A single-channel encoder does not provide any direction information and does not support bidirectional fluid metering.

The dual counter mode supports two independent single-channel meters. Each meter has its own independent counter and digital comparator. Some of the global configuration settings apply to both channels, such as pull-up current, sampling rate, and debounce time. The dual mode may also be used for a redundant count using a two-channel non-quadrature encoder.

Quadrature counter mode supports a single two-channel quadrature meter encoder. The quadrature counter mode supports bidirectional encoders and applications with bidirectional fluid flow. In quadrature counter mode, clock-wise counts will increment counter 0, while counter clock-wise counts will increment counter 1. Subtracting counter 1 from counter 0 will yield the net position. If the normal fluid flow is clockwise, then the counter clockwise counter 1 value represents the cumulative back-flow. Firmware may use the back-flow counter with the corresponding comparator to implement a back-flow alarm. The clock-wise sequence is (LL-HL-HH-LH), and the counter clock-wise sequence is (LL-LH-HH-HL). (For this sequence LH means Channel 1 = Low and Channel 0 = High.)

Firmware cannot write to the counters. The counters are reset when the CONFIG register is written and have their counting enabled when the PCMD field is set to either single, dual, or quadrature modes. The counters only increment and will roll over to 0x000000 after reaching 0xFFFFFF.

For single mode, the channel 0 input connects to counter 0. In dual mode, the channel 0 input connects to counter 0 while the channel 1 input connects to counter 1. In Quadrature mode, clock-wise counts are sent to counter 0 while counter clock-wise counts are sent to counter 1.

14.7.1. Quadrature Error

The quadrature encoder must only send valid quadrature codes. A valid quadrature sequence consists of four valid states. The quadrature codes are only permitted to transition to one of the adjacent states, and an invalid transition will result in a quadrature error. Note that a quadrature error is likely to occur when first enabling the quadrature counter mode, since the Advanced Capture Counter state machine starts at the LL state and the initial state of the quadrature is arbitrary. It is safe to ignore the first quadrature error immediately after initialization.

14.7.2. Flutter Detection

The flutter detection logic can be used with either quadrature counter mode or dual counter mode when the two channels are expected to be in step. Flutter refers to the case where one input continues toggling while the other input stops toggling. This may indicate a broken switch or a pressure oscillation when the wheel magnet stops at just the right location. If a pressure oscillation causes a slight rotational oscillation in the wheel, it could cause a number of pulses on one of the inputs, but not on the other. All four edges are checked by the flutter detection feature (Channel 1 positive, Channel 1 negative, Channel 0 positive, and Channel 0 negative).When enabled, Flutter detection may be used as an interrupt or wake-up source.

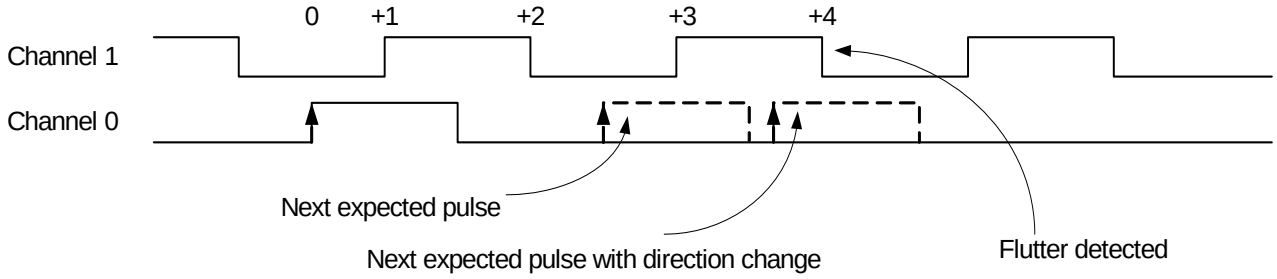


Figure 14.5. Flutter Example

For example, flutter detected on the channel 0 positive edge means that 4 edges (positive or negative) were detected on channel 1 since the last channel 0 positive edge. Each channel 0 positive edge resets the flutter detection counter while either channel 1 edge increments the counter. There are similar counters for all four edges.

The flutter detection circuit provides interrupts or wake-up sources, but firmware must also read the Advanced Capture Counter registers to determine what corrective action, if any, must be taken.

On the start of flutter event, the firmware should save both counter values and the DIRHIST field in the STATUS register. Once the end of flutter event occurs the firmware should also save both counter values and the DIRHIST field. The flutter stop enable bit (FLSTPEN in CONFIG) may be set to stop the counters when flutter is occurring (quadrature mode only). For quadrature mode, the opposite counter should be decremented by one. In other words, if the direction was clock-wise, the counter clock-wise counter (counter 1) should be decremented by one to correct for one increment before flutter was detected. For dual mode, two switches can be used to get a redundant count. If flutter starts during dual mode, both counters should be saved by firmware. After flutter stops, both counters should be read again. The counter that incremented the most was the one that picked up the flutter. There is also a mode to switch from quadrature to dual when flutter occurs, using the FLQDEN bit in CONFIG. This changes the counter style from quadrature (count on any edge of channel 1 or channel 0) to dual to allow all counts to be recorded. Once flutter ends, this mode switches the counters back to quadrature mode. Flutter stop enable does not function when the FLQDEN bit is set.

14.8. Sample Rate

The Advanced Capture Counter has a programmable sampling rate. The Advanced Capture Counter initiates samples at discrete time intervals based RTC0TCLK cycles. The PERIOD field sets the sampling rate. The system designer should carefully consider the maximum pulse rate for the particular application when setting the sampling rate and debounce time. Sample rates from 4 to 4096 RTC0TCLK cycles can be selected to either further reduce power consumption or work with shorter pulse widths. The slowest sampling rate will provide the lowest possible power consumption.

14.9. Debounce

Many types of mechanical switches exhibit switch bouncing that could potentially result in false counts or quadrature errors. The Advanced Capture Counter includes digital debounce logic using a digital integrator that can eliminate false counts due to switch bounce. In switch topology mode, the input of the integrator connects to the IN0 and IN1 inputs directly. In LC resonant mode, the input of the integrator is fed by the output of the discriminator logic. The output of the integrator connects to the counters.

The debounce integrator has two independent programmable thresholds: one for the rising edge and one for the falling edge. The HDBTH field in the DBCONFIG register sets the threshold for the rising edge (high debounce). This field sets the number of cumulative high samples required to output a logic high to the counter. The LDBTH field in DBCONFIG sets the threshold for the falling edge (low debounce). This field sets the number of cumulative low samples required to output a logic low to the counter.

Note that the debounce integrator does not count consecutive samples. Requiring consecutive samples would be susceptible to noise. The digital integrator inherently filters out noise.

The system designer should carefully consider the maximum anticipated counter frequency and duty-cycle when setting the debounce time. If the debounce configuration is set too large, the Advanced Capture Counter will not count short pulses. The debounce-high configuration should be set to less than one-half the minimum input pulse high-time. Similarly, the debounce-low configuration should be set to less than one-half the minimum input pulse low-time.

Figure 14.6 illustrates the operation of the debounce integrator when used in conjunction with a switch topology circuit. The top waveform is the representation of the reed switch (high: open, low: closed) which shows some random switch bounce. The bottom waveform is the final signal that goes into the counter which has the switch bounce removed. Based on the actual reed switch used and sample rate, the switch bounce time may appear shorter in duration than the example shown here. The second waveform is the pull-up resistor enable signal. The enable signal enables the pull-up resistor when high and disables when low. ACCTR0_IN0 is the line to the reed switch. On the right side of ACCTR0_IN0 waveform, the line voltage is decreasing towards ground when the pull-up resistors are disabled. Beneath the charging waveform, the arrows represent the sample points. The Advanced Capture Counter samples the ACCTR0_IN0 voltage once the charging completes. The sensed ones and zeros are the sampled data. Finally the integrator waveform illustrates the output of the digital integrator. The integrator is set to 4 initially and counts to down to 0 before toggling the output low. Once the integrator reaches the low state, it needs to count up to 4 before toggling its output to the high state. The debounce logic filters out switch bounce or noise that appears for a short duration.

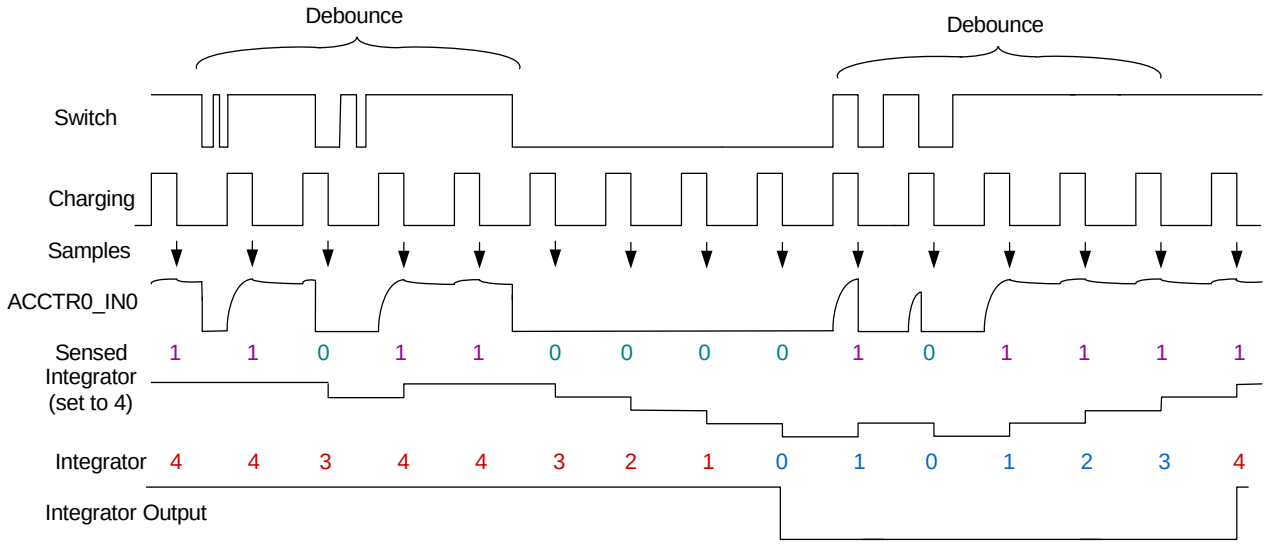


Figure 14.6. Debounce Timing

14.10. Reset Behavior

Unlike most MCU peripherals, an MCU reset does not completely reset the Advanced Capture Counter. This includes a power on reset and all other reset sources. An MCU reset does not clear the counter values. The Advanced Capture Counter registers do not reset to a default value except upon a power-on reset. The 24-bit counter values are persistent unless cleared manually by writing to the CONFIG register. Note that if the VBAT voltage ever drops below the minimum operating voltage, this may compromise contents of the counters.

The CONFIG register should normally be written only once after reset. This register sets the counter mode and resets the counter values when written.

Firmware should read the reset sources registers to determine the source of the last reset and initialize the Advanced Capture Counter accordingly.

When the Advanced Capture Counter resets, it takes some time (typically two RTC clock cycles) to synchronize between internal clock domains. The counters do not increment during this synchronization time.

14.11. Wake up and Interrupt Sources

The Advanced Capture Counter has multiple interrupt and wake-up source conditions. To enable an interrupt, enable the source using the STATUS register and enable the Advanced Capture Counter interrupt in the NVIC. The Advanced Capture Counter interrupt service routine should read the interrupt flags in STATUS to determine the source of the interrupt and clear the interrupt flags.

To enable the Advanced Capture Counter as a wake up source, enable the source in the STATUS and enable the Advanced Capture Counter as a wake-up source in the PMU module. Upon waking, firmware should read the PMU registers to determine the wake-up source. If the Advanced Capture Counter has woken the MCU, firmware should read the flag bits in STATUS to determine the Advanced Capture Counter wake-up source and clear the flag bits before going back to sleep. STATUS includes all interrupt and wake-up sources for the module.

14.12. Register Write Access

Writing to the ACCTR registers is synchronized to the RTC0TCLK domain. For this reason, some register writes may take many cycles before the actual register is updated. If a new write to a register is initiated before the previous write has completed, only the latest write may take. For this reason, the UPDTSTSF flag in the CONFIG register should be polled between consecutive writes, to ensure proper register updates.

14.13. Debug Signals

The Advanced Capture Counter module supports two output pins for hardware debug purposes (ACCTR0_DBG0 and ACCTR0_DBG1). The debug pins can be connected to the outputs of the comparators or integrators in either the LC or switch modes. To use the debug pins, the desired outputs should be selected with the DBGSEL field in the CONFIG register, and enabled using the DBGOEN in the DEBUGEN register.

14.14. LC Resonant Setup Example

This section details the basic steps to configure the pulse counter in LC resonant mode for an ac coupled circuit. There are a variety of topologies which can be used, but a simple tank circuit will be described here. For this example a 10 µH inductor is connected in parallel to a 0.01 µF capacitor. One side is grounded while the other side connects to a PMOS transistor and also to a 0.01 µF capacitor for ac coupling to the ACCTR0_LCIN0 pin. See Figure 14.7.

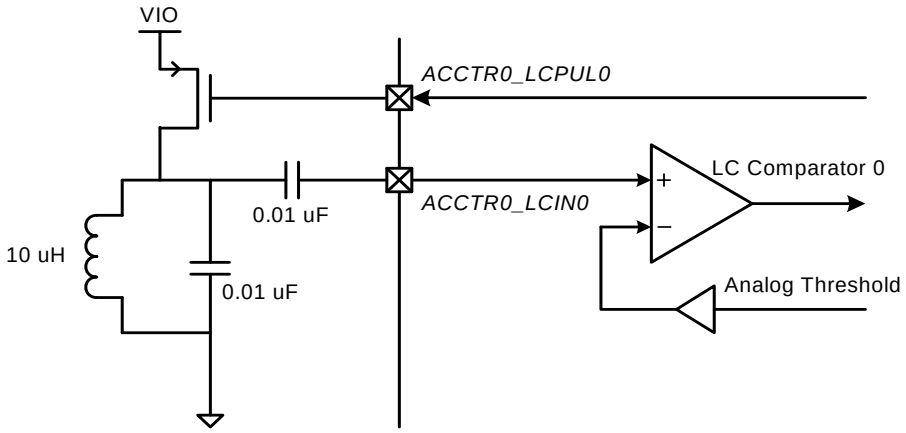


Figure 14.7. LC Resonant Example Schematic

The inductor must have a very low resistance in the tens of milliohms range. Even for an inductor with hundreds of milliohms, the dampening rate of the sine wave will be too fast for a useful circuit. The following steps outline the firmware setup to implement this LC resonant topology. The sequence is not unique to channel 0, and may be used to calibrate both channels at once in quadrature or dual mode.

1. Set the LCMD field to 0x4 to get a single ended, timed excitation pulse mode of operation.
2. Set the excitation pulse to be pulse low (PMD = 0x2) and set the bias pulses for internal use.
3. Set the bias timing offset to ½ cycle if needed. This gives ½ cycle of spacing between the bias pulse and the excitation pulse.
4. Set zone P to be 2 or 3 cycles long (3 cycles if using ½ cycle bias timing offset).
5. If using consecutive mode, set zone D to be the same number of cycles as zone P.
6. Set the PERIOD to 32 cycles or more which allows for non activity between samples and saves power
7. Set the comparator threshold to full range (CMP0THR = 1) and a low coarse threshold (about 0x04 in CMP0CTH) for the initial calibration measurement.
8. Set the comparator low side hysteresis (CMPLHYS) to 0V and the high side hysteresis (COMHHYS) to 10mV.
9. Select the fastest comparator speed setting (CMPMD = 0x3) if the LC resonant frequency is in the 500kHz to 1MHz range. For slower resonant frequencies the comparator speed setting can be lowered to save power.
10. Set the DBGOEN to 1 and write 0x2 to DBGSEL, and monitor both the ACCTR0_LCIN0 pin (comparator input) and the ACCTR0_DEBUG0 pin (comparator output) with an oscilloscope.
11. Perform an LC oscillator calibration by setting CLKCAL to 1. This will capture the number of LC oscillator cycles during a single RTC0TCLK period to the CLKCYCLES field. This number can then be divided to get a starting excitation pulse width for tuning the circuit. For example if the LC resonant frequency is 1MHz, using a excitation pulse wider than 1us would actually leave the external excitation transistor on for too long and begin lowering the final energy going into the LC resonant circuit as well as wasting power. As the LC resonant circuit swings back the opposite direction it will be fighting against additional power coming through the external excitation transistor. As an example, for an oscillator that is calibrated and reports

0x52B cycles per RTC0TCLK, the desired excitation pulse width of 1us can be obtained by dividing 0x52B by 30.5 µs per RTC0TCLK period which gives 0x2B for the timer RELOAD value. This number still does not guarantee that the ringing seen at the ACCTR0_LCIN0 pin will not be too great and cause the ESD protection to turn on and clip the incoming signal. The ac coupling capacitor should be chosen to get a reasonable size input voltage swing.

12. For the example LC resonant circuit the voltage will be swinging above and below 0V while the ACCTR0_LCIN0 pin on the opposite side of the ac coupling capacitor will be swinging around VBAT/2. If the voltage swing at ACCTR0_LCIN0 goes much below 0 V or above 5 V, the ESD protection circuitry will turn on momentarily and inject or remove charge resulting in an unwanted dc offset. To prevent this dc offset, use the excitation calibration circuit.
13. Power VBAT at 3.6 V.
14. Set calibration to run until pass, calibration on, and set the START bit to start the sequencer.
15. The result of the calibration should find an excitation pulse width that is wide enough to produce a good waveform but not so wide that it would cause ESD clipping. This calibration only generates the 5 LSBs of the 12-bit RELOAD field. It is assumed that the user has provided the 7 MSBs to the RELOAD field from the oscillator calibration step described in Step 11.
16. The calibration result is presented in the PUVAL field. These 5 bits should be written into RELOAD[4:0], leaving the 7 MSBs intact.
17. Monitor the waveform on an oscilloscope at ACCTR0_LCIN0 to double-check that there is no dc offset added by clipping.
18. Set the coarse threshold (CMP0CTH) to a mid level such as 0x20 for normal operation.
19. The integrator/debouncer settings can be used to filter some noise out of the system but integration also increases latency. More samples are needed before the integrator recognizes a new high or low value. To begin with, set the integrator debounce high and low settings to a low value such as 0x02. This can be increased if needed in a noisy environment.
20. There is also digital hysteresis for the discriminator that can be used to when the wheel stops at a location that produces readings close to the discriminator value. However, there needs to be enough headroom to use the digital hysteresis at low supply. In other words having a maximum LCCOUNT0 value of 0x09 and a minimum LCCOUNT0 value of 0x06 at 1.8 V would not allow enough headroom to set the digital hysteresis to a value of 0x03. Initially leave the digital hysteresis at a value of 0x00 or 0x01 unless the system is very noisy.
21. Disable the automatic tracking mode and the center discriminator mode (ACDEN and ATRKEN = 0).
22. Power VBAT at 1.8V.
23. Next, read the LCLIMITS register to reset the MAX0 to 0x00 and the MIN0 to 0xFF.
24. Run the wheel several times to capture the MAX and MIN values at 1.8 V.
25. Enable automatic tracking mode (ATRKEN = 1) and automatic center discriminator mode (ACDEN = 1). This will lock in a small window size for 1.8 V where the delta between MAX0 and MIN0 is the smallest due to the low supply. Had this window been locked in at 3.6 V, the tracking window would be too large to work well when the voltage dropped to 1.8 V.
26. As the VBAT voltage varies between 1.8 and 3.6 V the tracking window will move up and down to follow the maximum and minimum LCCOUNT values. These LCCOUNT values can vary quite a bit with voltage, and the tracking mechanism allows the pulse counter to keep the discriminator centered as the supply voltage changes. In quadrature mode, this maintains a good duty cycle with overlap between the two inputs. Without the overlap, the quadrature method used for directional counting would only see 3 states instead of 4 states.

14.15. ACCTR0 Registers

This section contains the detailed register descriptions for ACCTR0 registers.

Register 14.1. ACCTR0_CONFIG: Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PCMD		TOPMD	Reserved			FLSTPEN	FLQDEN	Reserved							
Type	RW		RW	R			RW	RW	R							
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								DBGSEL			Reserved				UPDSTSF
Type	R						RW		RW			R				R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_CONFIG = 0x4004_2000

Table 14.7. ACCTR0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:30	PCMD	**Pulse Counter Mode.** 00: Disable the pulse counter. 01: Select single channel mode. 10: Select dual channel mode. 11: Select quadrature mode.
29	TOPMD	**Topology Mode.** 0: Select the switch closure topology. 1: Select the LC resonant topology.
28:26	Reserved	Must write reset value.
25	FLSTPEN	**Flutter Stop Enable.** 0: The pulse counter continues operating during a flutter event. 1: The 24-bit counters stop counting during a flutter event.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.7. ACCTR0_CONFIG Register Bit Descriptions

Bit	Name	Function
24	FLQDEN	**Flutter Quadrature-to-Dual Switch Enable.** 0: The pulse counter remains in quadrature mode during a flutter event. 1: The pulse counter switches from quadrature mode to dual mode during a flutter event.
23:8	Reserved	Must write reset value.
7:5	DBGSEL	**Debug Signal Select.** This field selects which signals are sent to the debug output pins when DBGOEN is enabled. 000: No debug signals output. 001: (LC Mode) DBG0 = CMP0OUT, DBG1 = CMP1OUT. 010: (LC Mode) DBG0 = CMP0OUT, DBG1 = INTEG0. 011: (LC Mode) DBG0 = CMP1OUT, DBG1 = INTEG1. 100: (Any Mode) DBG0 = INTEG0 DBG1 = INTEG1. 101: (Switch Mode) DBG0 = CMP0OUT, DBG1 = CMP1OUT. 110: (Switch Mode) DBG0 = CMP0OUT, DBG1 = INTEG0. 111: (Switch Mode) DBG0= CMP1OUT, DBG1 = INTEG1.
4:1	Reserved	Must write reset value.
0	UPDSTSF	**Write Update Status Flag.** This flag indicates that an update to an internal pulse counter register is in progress. Firmware must wait to do any additional updates to the pulse counter registers until hardware clears this bit. 0: An internal pulse counter register update is not in progress. 1: An internal pulse counter register update is in progress.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.2. ACCTR0_CONTROL: Control Register

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CALBUSYF	CALRF	CALSEL	PUVAL					FPDNEN	FPUPEN	CALPUMD		CALMD	CMPHTH		CMPLTH[1]
Type	RW	R	RW	RW					RW	RW	RW		RW	RW		RW
Reset	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CMPLTH[0]	Reserved														
Type	RW	R														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_CONTROL = 0x4004_2010

Table 14.8. ACCTR0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	CALBUSYF	**Calibration Busy Flag.** 0: A calibration operation is not in progress. 1: A calibration operation is in progress. Hardware will clear this flag when the operation completes.
30	CALRF	**Calibration Result Flag.** 0: The automatic calibration operation did not succeed. 1: The automatic calibration operation succeeded.
29	CALSEL	**Automatic Calibration Input Select.** 0: Calibrate the IN0 input. 1: Calibrate the IN1 input.
28:24	PUVAL	**Pull-up Value.** When operating with a switch topology (TOPMD = 0), bits [4:2] of this field are the pull-up value and bits [1:0] are the duty cycle. Writing a value of 0 to this field disables the pull-ups. When operating in LC mode (TOPMD = 1), this field contains the lowest 5 bits of the oscillator counter.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.8. ACCTR0_CONTROL Register Bit Descriptions

Bit	Name	Function
23	FPDNEN	**Force Ground Input Enable.** This bit setting overrides the FPUPEN setting. 0: Disable input grounding. 1: Enable input grounding. The IN0 and IN1 inputs are grounded.
22	FPUPEN	**Force Continuous Pull-up Enable.** 0: Pull-ups are enabled automatically by hardware. 1: Always enable the pull-ups.
21:20	CALPUMD	**Automatic Calibration Pull-up Mode.** 00: Use full pull-up mode. 01: Use small pull-up mode. 10: Use medium pull-up mode. 11: Use large pull-up mode.
19	CALMD	**Automatic Calibration Mode.** 0: Continue to calibrate until a passing condition occurs. 1: Continue to calibrate until a failing condition occurs.
18:17	CMPHTH	**Comparator High Threshold.** 00: Set the digital comparator high threshold to 48% of VIO. 01: Set the digital comparator high threshold to 52% of VIO. 10: Set the digital comparator high threshold to 56% of VIO. 11: Set the digital comparator high threshold to 60% of VIO.
16:15	CMPLTH	**Comparator Low Threshold.** 00: Set the digital comparator low threshold to 32% of VIO. 01: Set the digital comparator low threshold to 36% of VIO. 10: Set the digital comparator low threshold to 40% of VIO. 11: Set the digital comparator low threshold to 44% of VIO.
14:0	Reserved	Must write reset value.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.3. ACCTR0_LCCONFIG: LC Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	FCMP1EN	FCMP0EN	CMP0CNT1EN	CMPMD		CMPHHYS		CMPLHYS		CMP1THR	CMP1CTH[5:1]				
Type	R	RW	RW	RW	RW		RW		RW		RW	RW				
Reset	0	0	0	0	0	0	0	1	0	1	1	1	0	0	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CMP1CTH[0]	CMP1FTH			CMP0THR	CMP0CTH						CMP0FTH			PEMD	
Type	RW	RW			RW	RW						RW			RW	
Reset	1	0	0	0	1	1	0	0	0	1	1	0	0	0	0	1

Register ALL Access Address

ACCTR0_LCCONFIG = 0x4004_2020

Table 14.9. ACCTR0_LCCONFIG Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	FCMP1EN	**Force LC Comparator 1 On Enable.** 0: Hardware automatically turns LC comparator 1 on and off. 1: Force LC comparator 1 always on.
29	FCMP0EN	**Force LC Comparator 0 On Enable.** 0: Hardware automatically turns LC comparator 0 on and off. 1: Force LC comparator 0 always on.
28	CMP0CNT1EN	**LC Comparator 0 to Count 1 Enable.** 0: Use LC comparator 0 as an input to counter 0 and LC comparator 1 as an input to counter 1. 1: Use LC comparator 0 as an input to both counter 0 and counter 1.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.9. ACCTR0_LCCONFIG Register Bit Descriptions

Bit	Name	Function
27:26	CMPMD	**LC Comparator Mode.** 00: Mode 0 (slowest response time, lowest power consumption). 01: Mode 1. 10: Mode 2. 11: Mode 3 (fastest response time, highest power consumption).
25:24	CMPHHYS	**LC Comparator High-side Hysteresis.** 00: Set both LC comparators to use 0 mV high-side hysteresis. 01: Set both LC comparators to use 5 mV high-side hysteresis. 10: Set both LC comparators to use 10 mV high-side hysteresis. 11: Set both LC comparators to use 20 mV high-side hysteresis.
23:22	CMPLHYS	**LC Comparator Low-side Hysteresis.** 00: Set both LC comparators to use 0 mV low-side hysteresis. 01: Set both LC comparators to use 5 mV low-side hysteresis. 10: Set both LC comparators to use 10 mV low-side hysteresis. 11: Set both LC comparators to use 20 mV low-side hysteresis.
21	CMP1THR	**LC Comparator 1 Threshold Range.** 0: Set the comparator 1 threshold to the low range (0 V to VIO/8 in 48 steps). 1: Set the comparator 1 threshold to a full range (0 V to VIO in 64 steps).
20:15	CMP1CTH	**LC Comparator 1 Coarse Threshold.** This field sets the LC comparator 1 coarse threshold. The valid coarse setting is 0 to 63 for full range mode and 0 to 7 for low range mode. In low range mode, this field value is set to the desired continuous setting divided by 6.
14:12	CMP1FTH	**LC Comparator 1 Fine Threshold.** This field sets the LC comparator 0 fine threshold. The valid fine threeshold setting is 0 to 5. The fine threshold is used only for low range. This field value is set to the modulus 6 of the desired continuous setting.
11	CMP0THR	**LC Comparator 0 Threshold Range.** 0: Set the comparator 0 threshold to the low range (0 V to VIO/8 in 48 steps). 1: Set the comparator 0 threshold to a full range (0 V to VIO in 64 steps).
10:5	CMP0CTH	**LC Comparator 0 Coarse Threshold.** This field sets the LC comparator 0 coarse threshold. The valid coarse setting is 0 to 63 for full range mode and 0 to 7 for low range mode. In low range mode, this field value is set to the desired continuous setting divided by 6.
4:2	CMP0FTH	**LC Comparator 0 Fine Threshold.** This field sets the LC comparator 0 fine threshold. The valid fine threeshold setting is 0 to 5. The fine threshold is used only for low range. This field value is set to the modulus 6 of the desired continuous setting.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.9. ACCTR0_LCCONFIG Register Bit Descriptions

Bit	Name	Function
1:0	PEMD	**LC Pulse Extension Mode.** 00: Stretch the LC comparator output low pulses by approximately 20 ns. 01: Stretch the LC comparator output high pulses by approximately 20 ns. 10: No pulse extension. 11: Reserved.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.4. ACCTR0_TIMING: Timing

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PERIOD				START	WAKEMD			ZONEP			ZONEA			ZONEB[2:1]	
Type	RW				RW	RW			RW			RW			RW	
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ZONEB[0]	ZONEC			ZONED			Reserved				B1OEN	B0OEN	STATE		
Type	RW	RW			RW			R				RW	RW	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Register ALL Access Address

ACCTR0_TIMING = 0x4004_2030

Table 14.10. ACCTR0_TIMING Register Bit Descriptions

Bit	Name	Function
31:28	PERIOD	**Pulse Counter Period.** This field is the period of the pulse counter in RTC clock cycles. The period consists of the sample time and rest time. 0000: Set the period to 4 RTC cycles. 0001: Set the period to 8 RTC cycles. 0010: Set the period to 16 RTC cycles. 0011: Set the period to 32 RTC cycles. 0100: Set the period to 64 RTC cycles. 0101: Set the period to 128 RTC cycles. 0110: Set the period to 256 RTC cycles. 0111: Set the period to 512 RTC cycles. 1000: Set the period to 1024 RTC cycles. 1001: Set the period to 2048 RTC cycles. 1010: Set the period to 4096 RTC cycles. 1011-1101: Reserved. 1110: Set the module to single sample mode and disable the period counter after the next completion of the sequencer. In this mode, firmware must start each sample by setting FLCSEN to 1. 1111: Set the module to consecutive sample mode and disable the period counter. After completing zone D, the timing engine will jump directly to zone A, skipping both the W and P zones.
27	START	**Sequencer Start.** This bit is used to start the sequencer. Hardware will automatically clear this bit to 0. 0: Do not start the sequencer. 1: Start the sequencer.
26:24	WAKEMD	**LC Wake Mode.** 000: Disable wake up events. 001: Wake or interrupt at the start of zone P. 010: Wake or interrupt at the start of zone A. 011: Wake or interrupt at the start of zone B. 100: Wake or interrupt at the start of zone C. 101: Wake or interrupt at the start of zone D. 110: Wake or interrupt at the end of the LC sequence. 111: Wake or interrupt at the end of the LC sequence and stop the sequencer when this event occurs.
23:21	ZONEP	**Zone P Count.** This field is the number of cycles + 1 in zone P.
20:18	ZONEA	**Zone A Count.** This field is the number of cycles + 1 in zone A.
17:15	ZONEB	**Zone B Count.** This field is the number of cycles + 1 in zone B.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.10. ACCTR0_TIMING Register Bit Descriptions

Bit	Name	Function
14:12	ZONEC	**Zone C Count.** This field is the number of cycles + 1 in zone C.
11:9	ZONED	**Zone D Count.** This field is the number of cycles + 1 in zone D.
8:5	Reserved	Must write reset value.
4	B1OEN	**Bias 1 Offset Enable.** 0: The bias 1 pulse is a full width (minimum 2 RTC cycles). 1: The bias 1 pulse is delayed 1/2 an RTC cycle and de-asserts 1/2 an RTC cycle early (minimum 3 RTC cycles).
3	B0OEN	**Bias 0 Offset Enable.** 0: The bias 0 pulse is a full width (minimum 2 RTC cycles). 1: The bias 0 pulse is delayed 1/2 an RTC cycle and de-asserts 1/2 an RTC cycle early (minimum 3 RTC cycles).
2:0	STATE	**Timing State.** This field is the current timing state of the pulse counter. Firmware should read this field twice to read the correct value.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.5. ACCTR0_LCMODE: LC Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LCMD				BMD		B1POL	B1ZONEPEN	B1ZONEAEN	B1ZONEBEN	B1ZONECEN	B0POL	B0ZONEPEN	B0ZONEAEN	B0ZONEBEN	B0ZONECEN
Type	RW				RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PMD		P1ZONE		P0ZONE		C1ZONE		C0ZONE		LCD1HYS		LCD0HYS		ACDEN	ATRKEN
Type	RW		RW		RW		RW		RW		RW		RW		RW	RW
Reset	1	0	1	0	0	1	1	1	0	1	0	0	0	0	1	0

Register ALL Access Address

ACCTR0_LCMODE = 0x4004_2040

Table 14.11. ACCTR0_LCMODE Register Bit Descriptions

Bit	Name	Function
31:28	LCMD	**LC Mode.** 0000: The LC pulse asserts throughout zone A or zone C with a single-ended comparator using the counter and discriminator. 0001: The LC pulse asserts throughout zone A or zone C with differential comparators using the counter and discriminator. 0010: The LC pulse asserts throughout zone A or zone C with a single-ended comparator sampling and holding at the end of the LC pulse. 0011: The LC pulse asserts throughout zone A or zone C with differential comparators sampling and holding at the end of the LC pulse. 0100: The LC pulse starts at the beginning of zone A or C and stops with the timer with a single-ended comparator using the counter and discriminator. 0101: The LC pulse starts at the beginning of zone A or C and stops with the timer with differential comparators using the counter and discriminator. 0110: The LC pulse starts at the beginning of zone A or C and stops with the timer with a single-ended comparator sampling and holding at the end of the LC pulse. 0111: The LC pulse starts at the beginning of zone A or C and stops with the timer with differential comparators sampling and holding at the end of the LC pulse. 1000: The LC pulse starts at beginning of zone A or C and stops with the rising edge of the external stop input (STOPx) with a single-ended comparator using the counter and discriminator. 1001: The LC pulse starts at beginning of zone A or C and stops with the falling edge of the external stop input (STOPx) with single-ended comparators using the counter and discriminator. 1010: The LC pulse starts at beginning of zone A or C and stops with the rising edge of the external stop input (STOPx) with a single-ended comparator sampling and holding at the end of the LC pulse. 1011: The LC pulse starts at beginning of zone A or C and stops with the falling edge of the external stop input (STOPx) with single-ended comparators sampling and holding at the end of the LC pulse. 1100: Do not generate a pulse with a single-ended comparator using the timer and discrimintor. 1101: Do not generate a pulse with differential comparators using the timer and discrimintor. 1110: Do not generate a pulse with a single-ended comparator sampling and holding at the end of the zone. 1111: Do not genreate a pulse with differential comparators sampling and holding at the end of the zone.
27:26	BMD	**Bias Mode.** 00: Disable the bias signals. 01: Use the bias signals externally only (LCBIAS0 and LCBIAS1 outputs). 10: Use the bias signals internally only. 11: Use the bias signals externally (LCBIAS0 and LCBIAS1 outputs) and internally.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.11. ACCTR0_LCMODE Register Bit Descriptions

Bit	Name	Function
25	B1POL	**Bias 1 Polarity.** 0: Set bias 1 to idle high, pulse low. 1: Set bias 1 to idle low, pulse high.
24	B1ZONEPEN	**Bias 1 Zone P Enable.** 0: Disable bias 1 during zone P. 1: Enable bias 1 during zone P.
23	B1ZONEAEN	**Bias 1 Zone A Enable.** 0: Disable bias 1 during zone A. 1: Enable bias 1 during zone A.
22	B1ZONEBEN	**Bias 1 Zone B Enable.** 0: Disable bias 1 during zone B. 1: Enable bias 1 during zone B.
21	B1ZONECEN	**Bias 1 Zone C Enable.** 0: Disable bias 1 during zone C. 1: Enable bias 1 during zone C.
20	B0POL	**Bias 0 Polarity.** 0: Set bias 0 to idle high, pulse low. 1: Set bias 0 to idle low, pulse high.
19	B0ZONEPEN	**Bias 0 Zone P Enable.** 0: Disable bias 0 during zone P. 1: Enable bias 0 during zone P.
18	B0ZONEAEN	**Bias 0 Zone A Enable.** 0: Disable bias 0 during zone A. 1: Enable bias 0 during zone A.
17	B0ZONEBEN	**Bias 0 Zone B Enable.** 0: Disable bias 0 during zone B. 1: Enable bias 0 during zone B.
16	B0ZONECEN	**Bias 0 Zone C Enable.** 0: Disable bias 0 during zone C. 1: Enable bias 0 during zone C.
15:14	PMD	**LC Pulse Mode.** 00: Disable pulse mode. 01: Toggle at the start of zone A or zone C. 10: Set the pulse mode to idle high, pulse low. 11: Set the pulse mode to idle low, pulse high.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.11. ACCTR0_LCMODE Register Bit Descriptions

Bit	Name	Function
13:12	P1ZONE	**Pulse 1 Active Zone Select.** 00: Disable the pulse 1 output (LCPUL1). 01: Select zone C only as the active zone for the pulse 1 output (LCPUL1). 10: Select zone A only as the active zone for the pulse 1 output (LCPUL1). 11: Select zones A and C as the active zones for the pulse 1 output (LCPUL1).
11:10	P0ZONE	**Pulse 0 Active Zone Select.** 00: Disable the pulse 0 output (LCPUL0). 01: Select zone C only as the active zone for the pulse 0 output (LCPUL0). 10: Select zone A only as the active zone for the pulse 0 output (LCPUL0). 11: Select zones A and C as the active zones for the pulse 0 output (LCPUL0).
9:8	C1ZONE	**Counter 1 Active Zone Select.** This setting is used in dual and quadrature modes only. 00: Select zone A as the active zone for counter 1 (LCIN1 input). 01: Select zone B as the active zone for counter 1 (LCIN1 input). 10: Select zone C as the active zone for counter 1 (LCIN1 input). 11: Select zone D as the active zone for counter 1 (LCIN1 input).
7:6	C0ZONE	**Counter 0 Active Zone Select.** 00: Select zone A as the active zone for counter 0 (LCIN0 input). 01: Select zone B as the active zone for counter 0 (LCIN0 input). 10: Select zone C as the active zone for counter 0 (LCIN0 input). 11: Select zone D as the active zone for counter 0 (LCIN0 input).
5:4	LCD1HYS	**LC Discriminator 1 Digital Hysterisis.** The value of this field sets the amount of digital hysterisis for the LC disciminator with inputs LCCOUNT1 and CD1. The hysterisis is applied only to the high-to-low transistion. In all modes, the low-to-high transition occurs if LCCOUNT1 is greater than or equal to CD1. 00: A high-to-low transition occurs if LCCOUNT1 is less than CD1. 01: A high-to-low transition occurs if LCCOUNT1 is less than CD1 - 1. 10: A high-to-low transition occurs if LCCOUNT1 is less than CD1 - 2. 11: A high-to-low transition occurs if LCCOUNT1 is less than CD1 - 3.
3:2	LCD0HYS	**LC Discriminator 0 Digital Hysterisis.** The value of this field sets the amount of digital hysterisis for the LC disciminator with inputs LCCOUNT0 and CD0. The hysterisis is applied only to the high-to-low transistion. In all modes, the low-to-high transition occurs if COUNT0 is greater than or equal to CD0. 00: A high-to-low transition occurs if LCCOUNT0 is less than CD0. 01: A high-to-low transition occurs if LCCOUNT0 is less than CD0 - 1. 10: A high-to-low transition occurs if LCCOUNT0 is less than CD0 - 2. 11: A high-to-low transition occurs if LCCOUNT0 is less than CD0 - 3.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Table 14.11. ACCTR0_LCMODE Register Bit Descriptions

Bit	Name	Function
1	ACDEN	**Automatic Center Discriminator Enable.** 0: Disable automatic center discriminator mode. Firmware must set the CD0 and CD1 fields. 1: Enable automatic center discriminator mode. Hardware will keep the CD0 and CD1 fields centered between MAX and MIN.
0	ATRKEN	**Automatic Tracking Enable.** When this bit is set to 1, reading the MAX and MIN fields will not cause them to reset. 0: Disable automatic tracking. 1: Enable automatic tracking. A new MAX value of any size will increase both the MAX and MIN by 1, and a new MIN value of any size will decrease both the MAX and MIN by 1.

Notes:
1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.6. ACCTR0_LCCLKCONTROL: LC Clock Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				RELOAD											
Type	R				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			CLKCAL	CLKCYCLES											
Type	R			RW	R											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_LCCLKCONTROL = 0x4004_2050

Table 14.12. ACCTR0_LCCLKCONTROL Register Bit Descriptions

Bit	Name	Function
31:28	Reserved	Must write reset value.
27:16	RELOAD	**LC Oscillator Reload Value.** This field is the starting value for the LC oscillator timer. The timer generates a pulse that is ~25ns per the number of cycles specified by RELOAD. The timer is reloaded with the RELOAD value as needed.
15:13	Reserved	Must write reset value.
12	CLKCAL	**LC Oscillator Calibration Start.** Setting this bit starts an automatic hardware calibration of the LC oscillator. Hardware will automatically clear this bit when the calibration operation completes. The calibration is performed "on-the-fly" and the time needed for calibration is added to the end of the next sample sequence. 0: A calibration operation is not in progress. 1: Start an oscillator calibration or a calibration operation is in progress.
11:0	CLKCYCLES	**LC Oscillator Clock Cycles.** This field contains the number of LC oscillator cycles from the last LC oscillator calibration. This field is the number of ~40 MHz clock cycles during one RTC clock period.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.7. ACCTR0_LCLIMITS: LC Counter Limits

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MAX1								MIN1							
Type	R								R							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MAX0								MIN0							
Type	R								R							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Register ALL Access Address

ACCTR0_LCLIMITS = 0x4004_2060

Table 14.13. ACCTR0_LCLIMITS Register Bit Descriptions

Bit	Name	Function
31:24	MAX1	**LC Counter 1 Maximum Value.** This field records the maximum value from LC counter 1. If the ATRKEN bit is cleared to 0, this field clears to all 0's after a read. If the ATRKEN bit is set to 1, this field is not cleared by a read, a new counter 1 minimum value will reduce the value of MIN1 and this field by 1, and a new counter 1 maximum value will increase the value of MIN1 and this field by 1.
23:16	MIN1	**LC Counter 1 Minimum Value.** This field records the minimum value from LC counter 1. If the ATRKEN bit is cleared to 0, this field sets to all 1's after a read. If the ATRKEN bit is set to 1, this field is not cleared by a read, a new counter 1 minimum value will reduce the value of this field and MAX1 by 1, and a new counter 1 maximum value will increase the value of this field and MAX1 by 1.
15:8	MAX0	**LC Counter 0 Maximum Value.** This field records the maximum value from LC counter 0. If the ATRKEN bit is cleared to 0, this field clears to all 0's after a read. If the ATRKEN bit is set to 1, this field is not cleared by a read, a new counter 0 minimum value will reduce the value of MIN0 and this field by 1, and a new counter 0 maximum value will increase the value of MIN0 and this field by 1.
7:0	MIN0	**LC Counter 0 Minimum Value.** This field records the minimum value from LC counter 0. If the ATRKEN bit is cleared to 0, this field sets to all 1's after a read. If the ATRKEN bit is set to 1, this field is not cleared by a read, a new counter 0 minimum value will reduce the value of this field and MAX0 by 1, and a new counter 0 maximum value will increase the value of this field and MAX0 by 1.

Register 14.8. ACCTR0_LCCOUNT: LC Counters

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CD1								LCCOUNT1							
Type	RW								R							
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CD0								LCCOUNT0							
Type	RW								R							
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_LCCOUNT = 0x4004_2070

Table 14.14. ACCTR0_LCCOUNT Register Bit Descriptions

Bit	Name	Function
31:24	CD1	**LC Counter 1 Discriminator.** This field contains the LC counter 1 discriminator value. This discriminator value determines whether the LC counter 1 result should be considered a logic 1 or logic 0.
23:16	LCCOUNT1	**LC Counter 1.** This field contains the most recent LC counter 1 value.
15:8	CD0	**LC Counter 0 Discriminator.** This field contains the LC counter 0 discriminator value. This discriminator value determines whether the LC counter 0 result should be considered a logic 1 or logic 0.
7:0	LCCOUNT0	**LC Counter 0.** This field contains the most recent LC counter 0 value.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.9. ACCTR0_DBCONFIG: Pulse Counter Debounce Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved													INTEG1	INTEG0	INTEGDCEN
Type	R													R	R	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HDBTH								LDBTH							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
ACCTR0_DBCONFIG = 0x4004_2080

Table 14.15. ACCTR0_DBCONFIG Register Bit Descriptions

Bit	Name	Function
31:19	Reserved	Must write reset value.
18	INTEG1	**PC Integrator 1 Output.** 0: The integrator 1 output is low. 1: The integrator 1 output is high.
17	INTEG0	**PC Integrator 0 Output.** 0: The integrator 0 output is low. 1: The integrator 0 output is high.
16	INTEGDCEN	**PC Integrator Disconnect Enable.** 0: Connect integrator to 24 bit counter state machine logic. 1: Disconnect the integrators from the IN0 and IN1 inputs.
15:8	HDBTH	**Integrator High Debounce.** This field is the number of high counts before the integrator recognizes a high. These counts do not need to be consecutive.
7:0	LDBTH	**Integrator Low Debounce.** This field is the number of low counts before the integrator recognizes a low. These counts do not need to be consecutive.

Register 14.10. ACCTR0_COUNT0: Pulse Counter 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								COUNT0[23:16]							
Type	R								R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COUNT0[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_COUNT0 = 0x4004_2090

Table 14.16. ACCTR0_COUNT0 Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:0	COUNT0	**Pulse Counter 0.** This field is the latest pulse counter 0 value. This field is the total number of counts for single and dual modes or clockwise counts for quadrature mode. Firmware must read all 24 bits of this field in one word read operation.

Notes:

1. The access methods for this register are restricted. Do not use half-word or byte access methods on this register.

Register 14.11. ACCTR0_COUNT1: Pulse Counter 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								COUNT1[23:16]							
Type	R								R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COUNT1[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_COUNT1 = 0x4004_20A0

Table 14.17. ACCTR0_COUNT1 Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:0	COUNT1	**Pulse Counter 1.** This field is the latest pulse counter 1 value. This field is the total number of counts for single and dual modes or counter-clockwise counts for quadrature mode. Firmware must read all 24 bits of this field in one word read operation.

Notes:

1. The access methods for this register are restricted. Do not use half-word or byte access methods on this register.

Register 14.12. ACCTR0_COMP0: Comparator 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								COMP0[23:16]							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COMP0[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ACCTR0_COMP0 = 0x4004_20B0

Table 14.18. ACCTR0_COMP0 Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:0	COMP0	**Pulse Counter Comparator 0 Threshold.** This field is PC counter 0's digital comparator threshold. Firmware must write all 24 bits of this field in one word write operation.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.13. ACCTR0_COMP1: Pulse Counter Comparator 1 Threshold

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								COMP1[23:16]							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COMP1[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ACCTR0_COMP1 = 0x4004_20C0

Table 14.19. ACCTR0_COMP1 Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:0	COMP1	**Pulse Counter Comparator 1 Threshold.** This field is PC counter 1's digital comparator threshold. Firmware must write all 24 bits of this field in one word write operation.

Notes:

1. Sequential writes to this register require polling on the update status flag (UPDSTSF) between each write

Register 14.14. ACCTR0_STATUS: Pulse Counter Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		CMP1OUT	CMP0OUT	DIRHIST				FLF	DIRF	STATE		IN1PREV	IN0PREV	IN1	IN0
Type	R		R	R	R				R	R	R		R	R	R	R
Reset	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	FLSTARTIEN	FLSTOPIEN	QERRIEN	TRANSIEN	CMP1IEN	CMP0IEN	OVFIEN	DIRCHGIEN	FLSTARTI	FLSTOPI	QERRI	TRANSI	CMP1I	CMP0I	OVFI	DIRCHGI
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ACCTR0_STATUS = 0x4004_20D0

Table 14.20. ACCTR0_STATUS Register Bit Descriptions

Bit	Name	Function
31:30	Reserved	Must write reset value.
29	CMP1OUT	**Comparator 1 Output.** This bit reports the value at the output of PC comparator 1 or LC comparator 1 depending on the selected mode of the module. 0: The output of comparator 1 is low. 1: The output of comparator1 is high.
28	CMP0OUT	**Comparator 0 Output.** This bit reports the value at the output of PC comparator 0 or LC comparator 0 depending on the selected mode of the module. 0: The output of comparator 0 is low. 1: The output of comparator 0 is high.
27:24	DIRHIST	**Direction History .** This bit field contains the last four recorded directions. A 1 represents clockwise and a 0 represents counter-clockwise. This bit field is valid in quadrature mode only.
23	FLF	**Flutter Detected Flag.** Hardware sets this bit to 1 if it detects flutter while operating in dual or quadrature mode. Flutter indicates a disparity between the number of negative or positive edges of IN1 and IN0.

Table 14.20. ACCTR0_STATUS Register Bit Descriptions

Bit	Name	Function
22	DIRF	**Direction Flag.** This flag indicates the current direction in quadrature mode. With an order of IN1:IN0, a counter-clockwise rotation order is L:L, L:H, H:H, H:L, and a clockwise rotation order is L:L, H:L, H:H, L:H. 0: The current direction is counter-clockwise. 1: The current direction is clockwise.
21:20	STATE	**Pulse Counter State.** This field is the internal state of the pulse counters. Note that this field is not synchronized and should be read twice to ensure a valid read.
19	IN1PREV	**Previous Integrator 1 Output.** 0: The previous integrator 1 output was low. 1: The previous integrator 1 output was high.
18	IN0PREV	**Previous Integrator 0 Output.** 0: The previous integrator 0 output was low. 1: The previous integrator 0 output was high.
17	IN1	**Integrator 1 Output.** 0: The integrator 1 output is low. 1: The integrator 1 output is high.
16	IN0	**Integrator 0 Output.** 0: The integrator 0 output is low. 1: The integrator 0 output is high.
15	FLSTARTIEN	**Flutter Start Interrupt Enable.** 0: Disable flutter detection start events as an interrupt or wake up source. 1: Enable flutter detection start events as an interrupt or wake up source.
14	FLSTOPIEN	**Flutter Stop Interrupt Enable.** 0: Disable flutter detection end events as an interrupt or wake up source. 1: Enable flutter detection end events as an interrupt or wake up source.
13	QERRIEN	**Quadrature Error Interrupt Enable.** 0: Disable quadrature error as an interrupt or wake up source. 1: Enable quadrature error as an interrupt or wake up source.
12	TRANSIEN	**Integrator Transition Interrupt Enable.** 0: Disable integrator transitions as an interrupt or wake up source. 1: Enable integrator transitions as an interrupt or wake up source.
11	CMP1IEN	**Digital Comparator 1 Interrupt Enable.** 0: Disable comparator 1 as an interrupt or wake up source. 1: Enable comparator 1 as an interrupt or wake up source.
10	CMP0IEN	**Digital Comparator 0 Interrupt Enable.** 0: Disable comparator 0 as an interrupt or wake up source. 1: Enable comparator 0 as an interrupt or wake up source.

Table 14.20. ACCTR0_STATUS Register Bit Descriptions

Bit	Name	Function
9	OVFIEN	**Counter Overflow Interrupt Enable.** 0: Disable counter overflows as an interrupt or wake up source. 1: Enable counter overflows as an interrupt or wake up source.
8	DIRCHGIEN	**Direction Change Interrupt Enable.** 0: Disable direction change as an interrupt or wake up source. 1: Enable direction change as an interrupt or wake up source.
7	FLSTARTI	**Flutter Start Interrupt Flag.** Hardware sets this flag to 1 when a flutter detection start event occurs. This bit is valid in dual or quadrature modes only. This bit must be cleared by firmware.
6	FLSTOPI	**Flutter Stop Interrupt Flag.** Hardware sets this flag to 1 when a flutter detection end event occurs. This bit is valid in dual or quadrature modes only. This bit must be cleared by firmware.
5	QERRI	**Quadrature Error Interrupt Flag.** Hardware sets this flag to 1 when a quadrature error occurs. This bit must be cleared by firmware.
4	TRANSI	**Integrator Transition Interrupt Flag.** Hardware sets this flag to 1 when either integrator output transitions from high-to-low or low-to-high. In LC mode, this bit indicates either that the P zone started or D zone completed based on the module configuration. This bit must be cleared by firmware.
3	CMP1I	**Digital Comparator 1 Interrupt Flag.** Hardware sets this flag to 1 when counter 1 equals the digital comparator 1 threshold. This bit must be cleared by firmware.
2	CMP0I	**Digital Comparator 0 Interrupt Flag.** Hardware sets this flag to 1 when counter 0 equals the digital comparator 0 threshold. This bit must be cleared by firmware.
1	OVFI	**Counter Overflow Interrupt Flag.** Hardware sets this flag to 1 to indicate that one of the PC counters overflowed. This bit must be cleared by firmware.
0	DIRCHGI	**Direction Change Interrupt Flag.** Hardware sets this flag to 1 to indicate a direction change. This flag is only valid in quadrature mode. This bit must be cleared by firmware.

Register 14.15. ACCTR0_DEBUGEN: Calibration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	DBGOEN	Reserved													
Type	R	RW	RW													
Reset	0	0	0	X	X	X	X	X	0	0	X	X	X	X	X	X

Register ALL Access Address

ACCTR0_DEBUGEN = 0x4004_20E0

Table 14.21. ACCTR0_DEBUGEN Register Bit Descriptions

Bit	Name	Function
31:15	Reserved	Must write reset value.
14	DBGOEN	**Debug Output Enable.** This bit enables the debug outputs on the DBG0 and DBG1 pins.
13:0	Reserved	Must write reset value.

14.16. ACCTR0 Register Memory Map

Table 14.22. ACCTR0 Memory Map

Register Name	ACCTR0_TIMING	ACCTR0_LCCONFIG	ACCTR0_CONTROL	ACCTR0_CONFIG
ALL Address	0x4004_2030	0x4004_2020	0x4004_2010	0x4004_2000
Access Methods	ALL	ALL	ALL	ALL
Bit 31	PERIOD	Reserved	CALBUSYF	PCMD
Bit 30		FCMP1EN	CALRF	
Bit 29		FCMP0EN	CALSEL	TOPMD
Bit 28		CMP0CNT1EN	PUVAL	Reserved
Bit 27	START	CMPMD		
Bit 26	WAKEMD	CMPHHYS		
Bit 25	ZONEP	CMPLHYS	FPDNEN	FLSTPEN
Bit 24		CMP1THR	FPUPEN	FLQDEN
Bit 23	ZONEA	CMP1CTH	CALPUMD	Reserved
Bit 22				
Bit 21				
Bit 20	ZONEB		CALMD	
Bit 19				
Bit 18			CMPHTH	
Bit 17				
Bit 16			CMPLTH	
Bit 15	ZONEC	CMP1FTH	Reserved	
Bit 14				
Bit 13				
Bit 12				
Bit 11	ZONED	CMP0THR		
Bit 10		CMP0CTH		
Bit 9				
Bit 8	Reserved			
Bit 7				DBGSEL
Bit 6				
Bit 5				
Bit 4	B1OEN	CMP0FTH		Reserved
Bit 3	B0OEN			
Bit 2	STATE	PEMD		
Bit 1				
Bit 0				UPDSTSF

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 14.22. ACCTR0 Memory Map

Register Name	ACCTR0_LCOUNT	ACCTR0_LCLIMITS	ACCTR0_LCCLKCONTROL	ACCTR0_LCMODE
ALL Address	0x4004_2070	0x4004_2060	0x4004_2050	0x4004_2040
Access Methods	ALL	ALL	ALL	ALL
Bit 31	CD1	MAX1	Reserved	LCMD
Bit 30				
Bit 29				
Bit 28				
Bit 27	LCCOUNT1	MIN1	RELOAD	BMD
Bit 26				
Bit 25				B1POL
Bit 24				B1ZONEPEN
Bit 23				B1ZONEAEN
Bit 22				B1ZONEBEN
Bit 21				B1ZONECEN
Bit 20				B0POL
Bit 19				B0ZONEPEN
Bit 18				B0ZONEAEN
Bit 17				B0ZONEBEN
Bit 16				B0ZONECEN
Bit 15	CD0	MAX0	Reserved	PMD
Bit 14				
Bit 13			CLKCAL	P1ZONE
Bit 12				
Bit 11	LCCOUNT0	MIN0	CLKCYCLES	P0ZONE
Bit 10				
Bit 9				C1ZONE
Bit 8				
Bit 7				C0ZONE
Bit 6				
Bit 5				LCD1HYS
Bit 4				
Bit 3				LCD0HYS
Bit 2				
Bit 1				ACDEN
Bit 0				ATRKEN

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 14.22. ACCTR0 Memory Map

Register Name	ACCTR0_DBCONFIG	ACCTR0_COUNT0	ACCTR0_COUNT1	ACCTR0_COMP0	ACCTR0_COMP1
ALL Address	0x4004_2080	0x4004_2090	0x4004_20A0	0x4004_20B0	0x4004_20C0
Access Methods	ALL	ALL	ALL	ALL	ALL
Bit 31	Reserved	Reserved	Reserved	Reserved	Reserved
Bit 30					
Bit 29					
Bit 28					
Bit 27					
Bit 26					
Bit 25					
Bit 24					
Bit 23					
Bit 22					
Bit 21					
Bit 20					
Bit 19					
Bit 18	INTEG1	COUNT0	COUNT1	COMP0	COMP1
Bit 17	INTEG0				
Bit 16	INTEGDCEN				
Bit 15	HDBTH				
Bit 14					
Bit 13					
Bit 12					
Bit 11					
Bit 10					
Bit 9					
Bit 8					
Bit 7	LDBTH				
Bit 6					
Bit 5					
Bit 4					
Bit 3					
Bit 2					
Bit 1					
Bit 0					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 14.22. ACCTR0 Memory Map

Register Name	ACCTR0_STATUS	ACCTR0_DEBUGEN
ALL Address	0x4004_20D0	0x4004_20E0
Access Methods	ALL	ALL
Bit 31	Reserved	Reserved
Bit 30		
Bit 29	CMP1OUT	
Bit 28	CMP0OUT	
Bit 27	DIRHIST	
Bit 26		
Bit 25		
Bit 24		
Bit 23	FLF	
Bit 22	DIRF	
Bit 21	STATE	
Bit 20		
Bit 19	IN1PREV	
Bit 18	IN0PREV	
Bit 17	IN1	
Bit 16	IN0	
Bit 15	FLSTARTIEN	
Bit 14	FLSTOPIEN	DBGOEN
Bit 13	QERRIEN	Reserved
Bit 12	TRANSIEN	
Bit 11	CMP1IEN	
Bit 10	CMP0IEN	
Bit 9	OVFIEN	
Bit 8	DIRCHGIEN	
Bit 7	FLSTARTI	
Bit 6	FLSTOPI	
Bit 5	QERRI	
Bit 4	TRANSI	
Bit 3	CMP1I	
Bit 2	CMP0I	
Bit 1	OVFI	
Bit 0	DIRCHGI	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

15. Advanced Encryption Standard (AES0)

This section describes the Advanced Encryption Standard (AES) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the AES block, which is used by all device families covered in this document.

15.1. AES Features

The AES module includes the following features:

- Operates on 4-word (16-byte) blocks.
- Supports key sizes of 128, 192, and 256 bits for both encryption and decryption.
- Generates the round key for decryption operations.
- All cipher operations can be performed without any firmware intervention for a set of 4-word blocks (up to 32 kB).
- Support for various chained and stream-ciphering configurations with XOR paths on both the input and output.
- Support for word, half-world, or byte access to DATA and XOR FIFO registers.
- Internal 4-word FIFOs to facilitate DMA operations.
- Integrated key storage.
- Hardware acceleration for Cipher-Block Chaining (CBC) and Counter (CTR) algorithms utilizing integrated counter-block generation and previous-block caching.

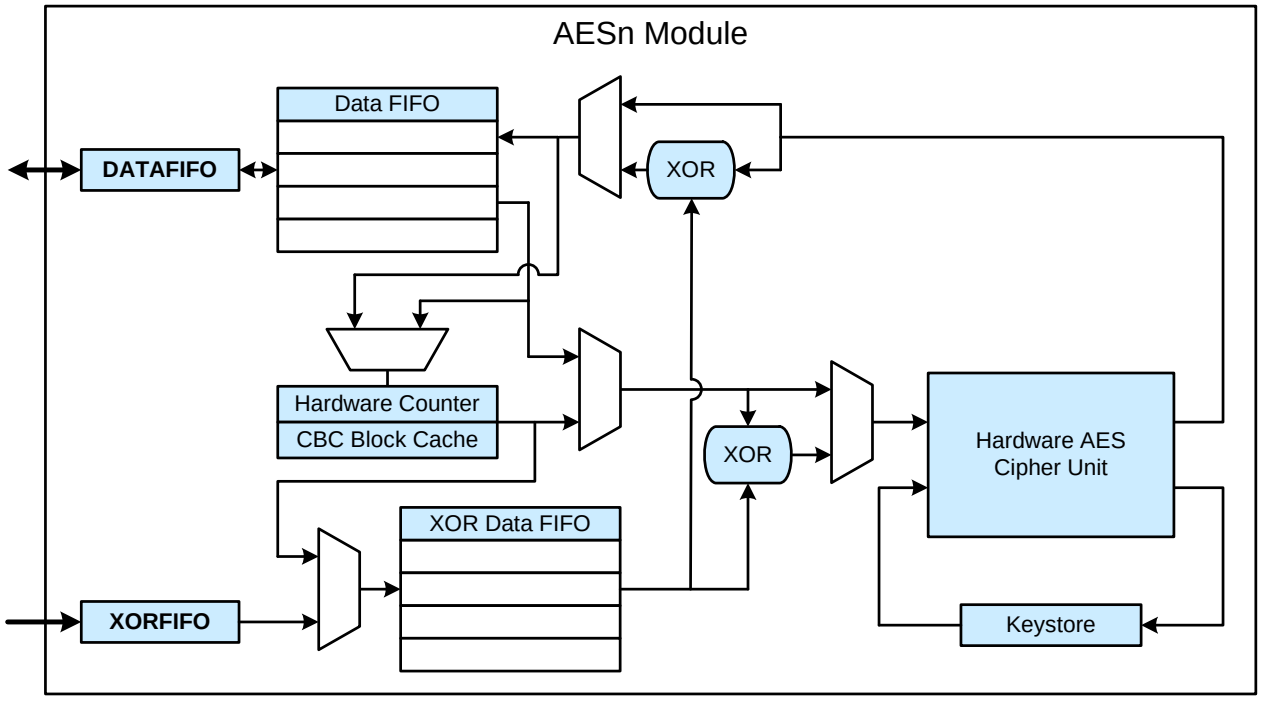


Figure 15.1. AES Block Diagram

15.2. Overview

The AES block cipher is a symmetric key encryption algorithm. Symmetric key encryption relies on secret keys that are known by both the sender and receiver. The decryption key may be obtained using a simple transformation of the encryption key. AES is not a public key encryption algorithm.

The AES block cipher uses a fixed 4-word (16-byte) block size. Data segments less than 4 words in length must be padded with zeros to fill the entire block.

Since symmetric key encryption relies on secret keys, the security of the data can only be protected if the key remains secret. If the encryption key is stored in Flash memory, then the entire Flash should be locked to ensure the encryption key cannot be discovered.

The hardware counter in the AES module can be programmed with an initial value using the HWCTR register. Similarly, the hardware keystore can be programmed with an initial value using the HWKEY register.

The basic AES block cipher is implemented in hardware. The integrated hardware acceleration for cipher block chaining (CBC) and counter (CTR) algorithms results in identical performance, memory bandwidth, and memory footprint between the most basic electronic codebook (ECB) algorithm and these more complex algorithms. This hardware accelerator provides performance that is much faster than a software implementation, which translates to more bandwidth available for other functions or a power savings for low-power applications.

15.2.1. Enabling the AES Module

Immediately following any device reset, the RESET bit is set to 1 to save power. All register bits except RESET are reset using the system reset or the RESET bit. To use the AES module, firmware must first clear the RESET bit before initializing the registers.

15.3. Interrupts

The AES interrupt flags are located in the STATUS register. The associated interrupt enable bits are located in the CONTROL register.

An AES completion interrupt can be generated if OCIEN is set to 1 whenever an encryption or decryption operation is complete. The completion interrupt should only be used in conjunction with software mode (SWMDEN bit is set to 1) and not with DMA operations, where the DMA completion interrupt should be used.

An AES error interrupt can be generated whenever an input/output data FIFO overrun (DORI = 1) or underrun (DURI = 1) error occurs, or when an XOR data FIFO overrun (XORI = 1) occurs. The error interrupts should always be enabled (ERRIEN = 1), even when using the DMA with the AES module.

15.4. Debug Mode

Firmware can clear the DBGMD bit to force the AES module to halt on a debug breakpoint. Setting the DBGMD bit forces the module to continue operating while the core halts in debug mode.

15.5. DMA Configuration and Usage

A DMA channel may be used to transfer data for the input/output data FIFO or the XOR data FIFO. The AES module FIFOs support word, half-word, or byte accesses. Each DMA transfer must consist of 4 words (16 bytes). To write to the input/output data FIFO, the DMA must move data from the source location in memory to the internal DATAFIFO register in non-incrementing mode. To read from the input/output data FIFO, the DMA must move data from the internal DATAFIFO register in non-incrementing mode to the destination location in memory. For the XOR data FIFO, the DMA must move data from the source location in memory to the internal XORFIFO register in non-incrementing mode. Firmware should only directly access the DATAFIFO and XORFIFO registers in software mode (SWMDEN bit is set to 1). AES FIFOs targeted by the DMA module should not be directly written to or read from.

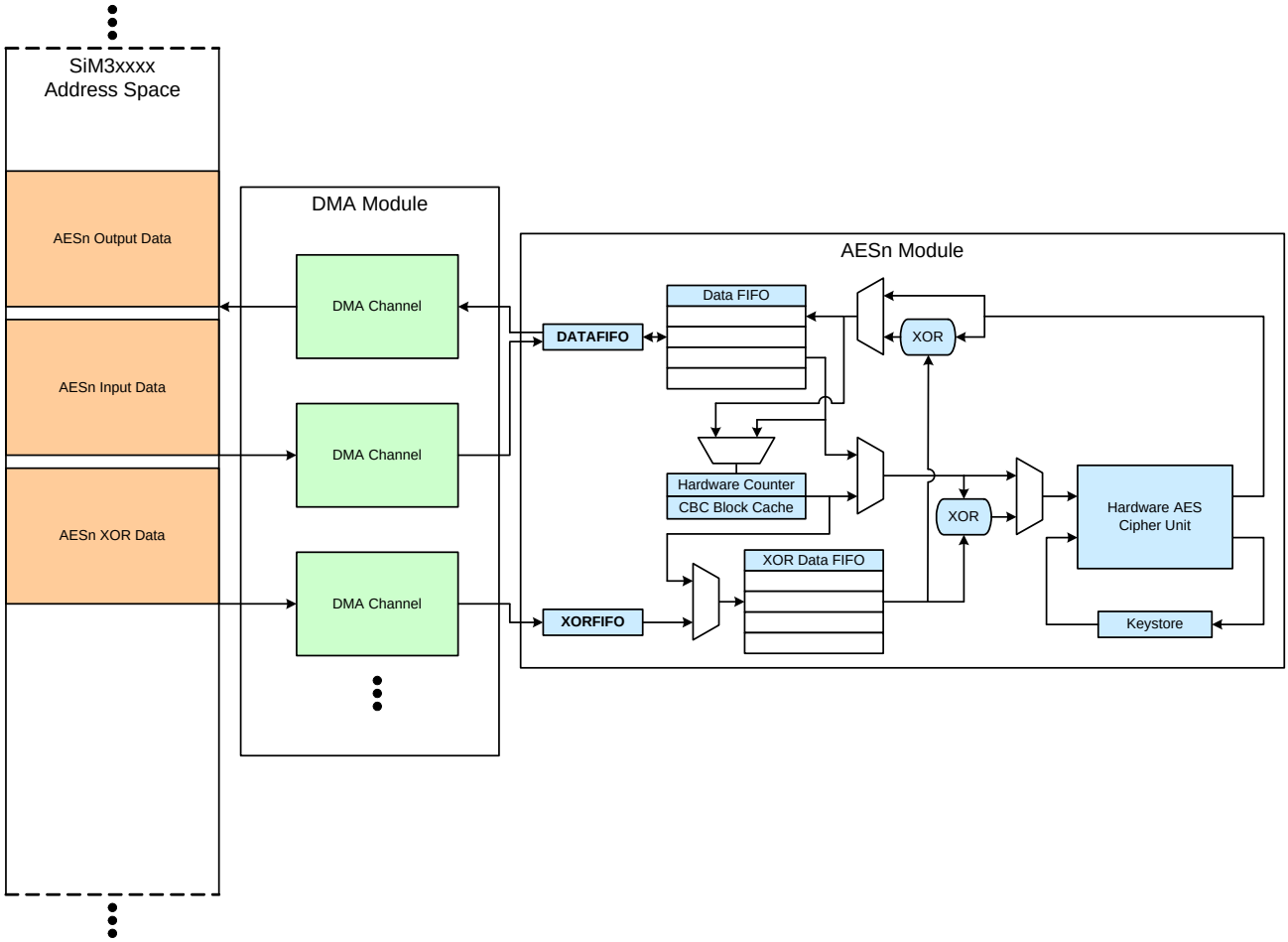


Figure 15.2. AES DMA Configuration

15.5.1. DMA and Interrupts

If a DMA channel is enabled for one of the FIFOs, the AES operation complete and error interrupts will not be suppressed. In software mode (SWMDEN = 1) the AES operation complete interrupt should be used, and in DMA mode (SWMDEN = 0) the DMA complete interrupt should be used. The error interrupt should be used in both software and DMA modes to notify the firmware of error conditions.

15.5.2. General DMA Transfer Setup

For the AES module, the DMA channels have these common settings:

- Source size (SRCSIZE) and destination size (DSTSIZE) set to 2 for word transfers, 1 for half-word transfers, or 0 for byte transfers. Each channel transfer size may be set independently.
- Number of transfers is (4 x N) – 1, where N is the number of 4-byte words.
- RPOWER set as indicated below, where the data transfer size equals 2^{RPOWER}:
  - For byte-accesses, RPOWER = 0,1,2,3 or 4
  - For half-word accesses, RPOWER = 0,1,2 or 3
  - For word accesses, RPOWER = 0,1 or 2
- The size of all memory buffers (input, output, and XOR) modulo 16 bytes must equal 0, so an even number of 4-word transfers must occur.

The input DMA channel should be programmed as follows:

- Destination end pointer set to the DATAFIFO register.
- Source end pointer set to the plain or cipher text input buffer address location + 16 x N – 4, where N is the number of blocks.
- The DSTAIMD field should be set to 11b for no increment.
- The SRCAIMD field should be set as indicated below:
  - For byte-accesses, SRCAIMD = 00b (the source address increments by one byte after each data transfer)
  - For half-word accesses, SRCAIMD = 01b, (the source address increments by one half-word after each data transfer)
  - For word accesses, SRCAIMD = 10b (the source address increments by one word after each data transfer)

The output DMA channel should be programmed as follows:

- Destination end pointer set to the plain or cipher text output buffer address location + 16 x N – 4, where N is the number of blocks.
- Source end pointer set to the DATAFIFO register.
- The DSTAIMD field should be set as indicated below:
  - For byte-accesses, DSTAIMD = 00b (the destination address increments by one byte after each data transfer)
  - For half-word accesses, DSTAIMD = 01b, (the destination address increments by one half-word after each data transfer)
  - For word accesses, DSTAIMD = 10b (the destination address increments by one word after each data transfer)
- The SRCAIMD field should be set to 11b for no increment.

The XOR DMA channel should be programmed as follows:

- Destination end pointer set to the XORFIFO register.
- Source end pointer set to the In XOR buffer address location + 16 x N – 4, where N is the number of blocks.
- The DSTAIMD field should be set to 11b for no increment.
- The SRCAIMD field should be set as indicated below:
  - For byte-accesses, SRCAIMD = 00b (the source address increments by one byte after each data transfer)
  - For half-word accesses, SRCAIMD = 01b, (the source address increments by one half-word after each data transfer)
  - For word accesses, SRCAIMD = 10b (the source address increments by one word after each data transfer)

To start a DMA operation with the AES module out of any device reset:

1. Set up the DMA channels for the input/output and XOR FIFOs depending on the desired cipher algorithm.
2. Clear the soft reset bit (RESET) to 0.
3. Configure the AES peripheral operation in the CONTROL, XFRSIZE, HWKEYx, and HWCTRx registers.
4. Start the AES operation by writing a 1 to XFRSTA.
5. Wait for the DMA completion interrupt.

15.6. Using the AES Module for Electronic Codebook (ECB)

The electronic codebook (ECB) cipher algorithm is the most basic block cipher mode since each cipher text output is only a function of its corresponding plain text input and the encryption key. This algorithm is shown in Figure 15.3.

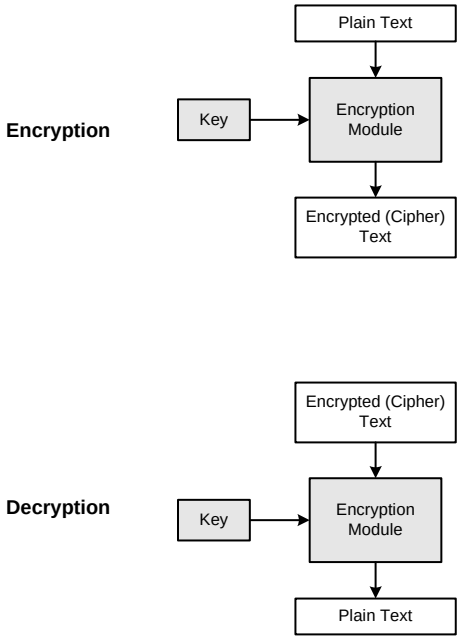


Figure 15.3. Electronic Codebook (ECB) Algorithm Diagram

The block diagram of the AES module performing this algorithm (encryption and decryption) is shown in Figure 15.4. The active data paths in this mode are shown in red.

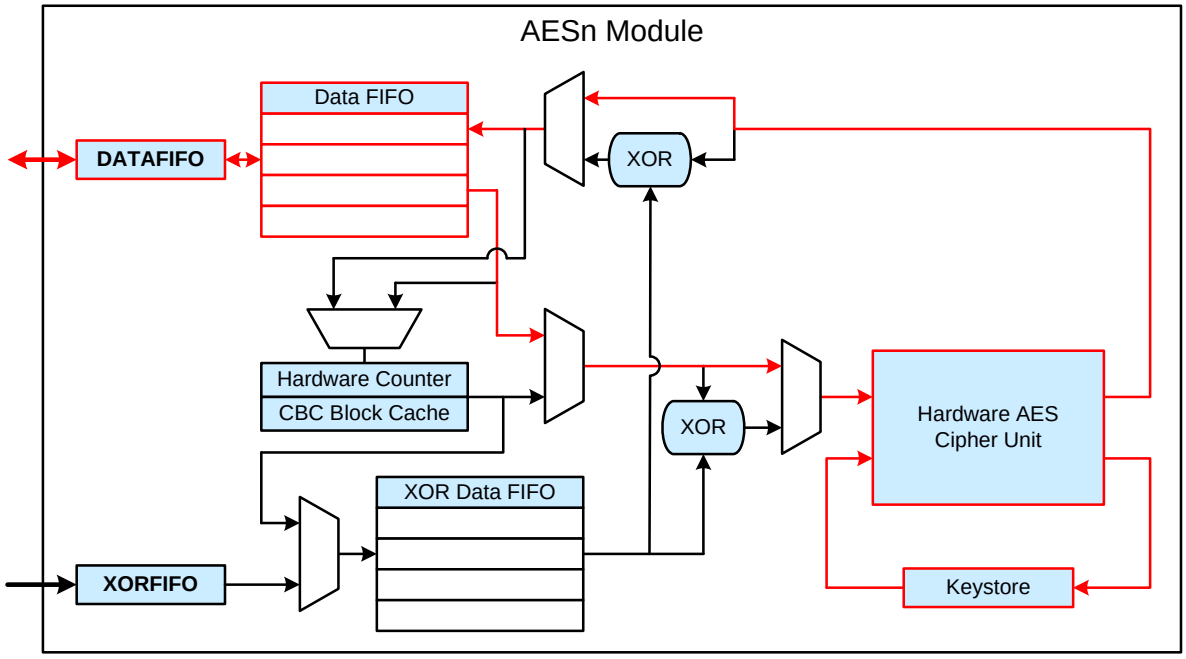


Figure 15.4. Electronic Codebook (ECB) AES Module Block Diagram—Encryption and Decryption

15.6.1. Configuring the DMA for ECB Encryption

To use the DMA with ECB encryption, the DMA and AES modules should be configured as follows:

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the plain text input buffer address location + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. EDMD set to 1 for encryption.
   d. KEYCPEN set to 1 to enable key capture at the end of the transaction.
   e. The HCBCEN, HCTREN, XOREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit. When the encryption process finishes, the decryption key will be available in the HWKEYx registers.

15.6.2. Configuring the DMA for ECB Decryption

To use the DMA with ECB decryption, the DMA and AES modules should be configured as follows:

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the cipher text input buffer address location + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the plain text output buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with decryption key value (automatically generated in the HWKEYx registers after the encryption process).
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. EDMD set to 1 for encryption.
   d. KEYCPEN set to 1 to enable key capture at the end of the transaction.
   e. The HCBCEN, HCTREN, XOREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit.

15.7. Using the AES Module for Cipher Block Chaining (CBC)

The cipher block chaining (CBC) cipher algorithm significantly improves the strength of basic ECB encryption by making each block encryption be a function of the previous block in addition to the current plain text and key. This algorithm is shown in Figure 15.5.

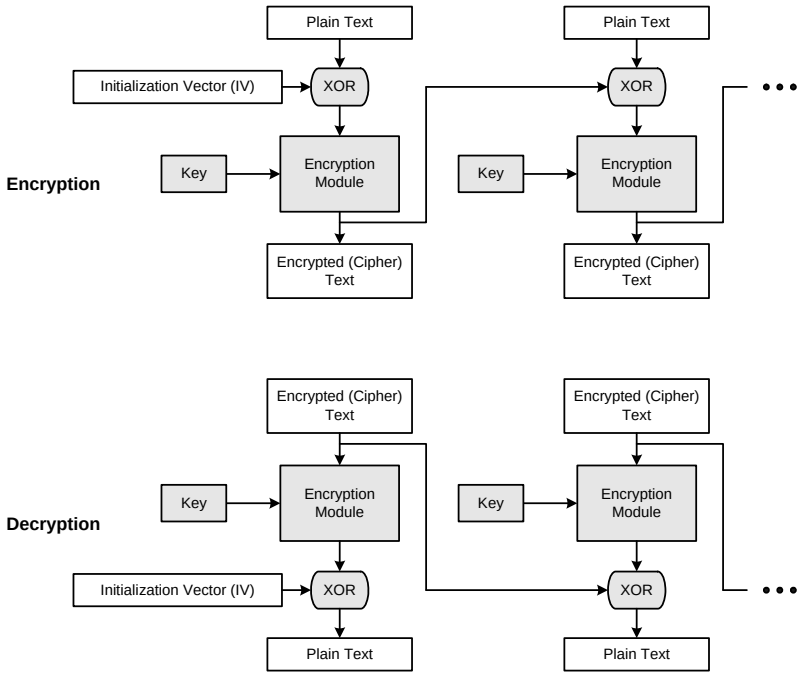


Figure 15.5. Cipher Block Chaining (CBC) Algorithm Diagram

The AES module can perform this algorithm by using a DMA channel or by using the AES hardware to feed the XOR data FIFO. Both of these methods are discussed.

15.7.1. Hardware XOR Channel Cipher Block Chaining

The AES module has a hardware chaining mode that accelerates the execution of the CBC algorithm. The module can reuse the hardware counter registers to temporarily store the previous result (CT(n-1)) for use with the next block. This eliminates the need for a third DMA channel, freeing it for other use, and reduces the timing to be identical with the counter (CTR) and much-simpler electronic codebook (ECB) algorithms. Any algorithms that use this hardware path, however, should not use the hardware counter (HCTREN set to 1) feature.

The block diagram of the AES module performing this encryption algorithm using the hardware path is shown in Figure 15.6. This decryption algorithm block diagram using the hardware path is shown in Figure 15.7. The active data paths in this mode are shown in red.

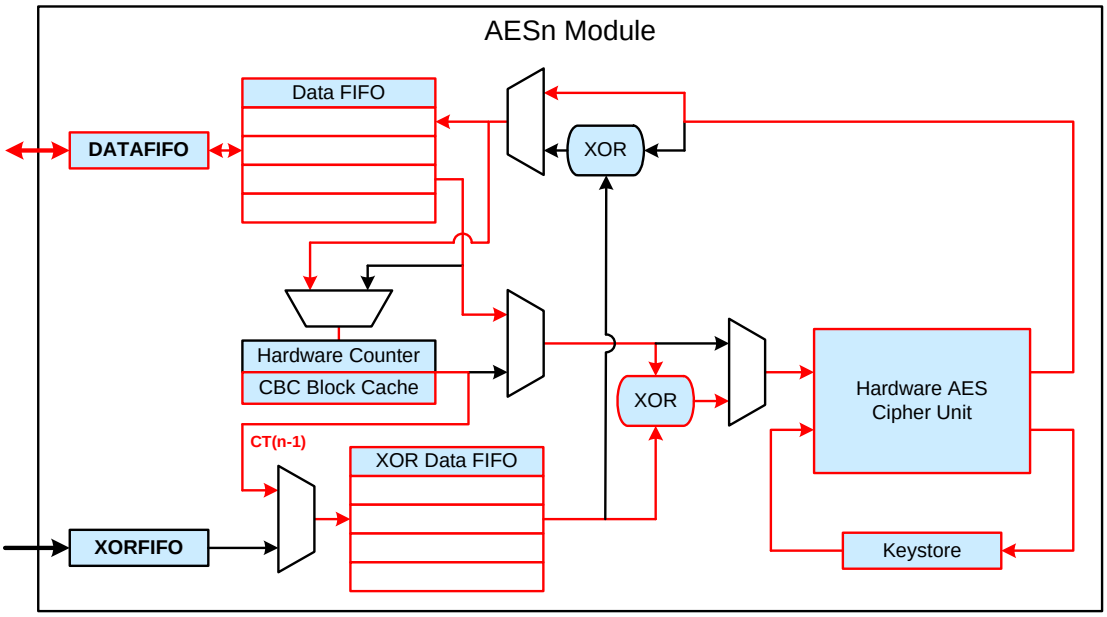


Figure 15.6. Hardware Cipher Block Chaining (CBC) AES Module Block Diagram—Encryption

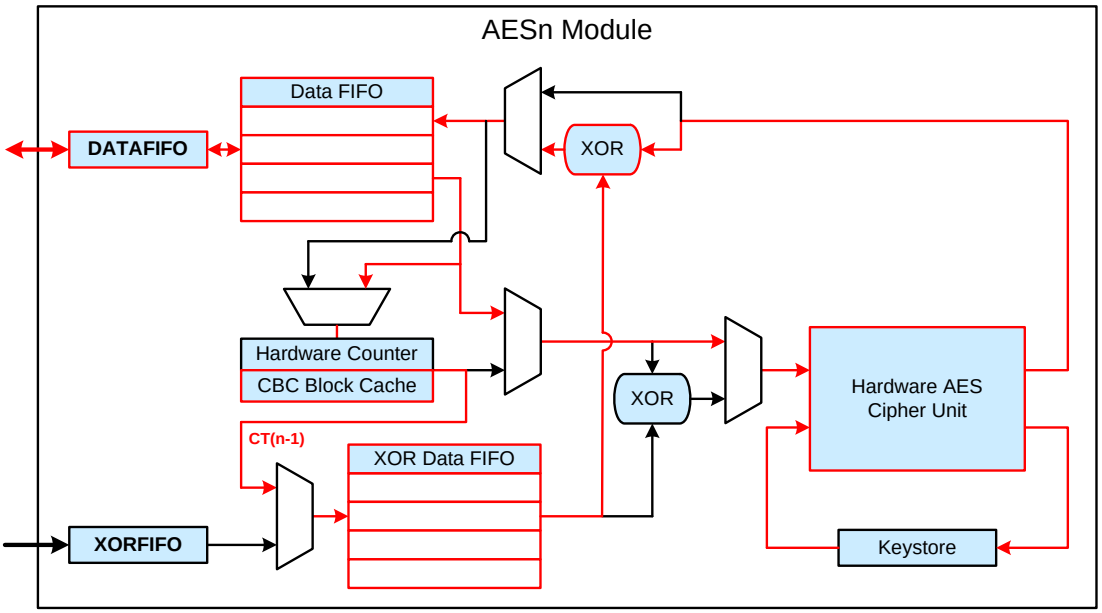


Figure 15.7. Hardware Cipher Block Chaining (CBC) AES Module Block Diagram—Decryption

15.7.1.1. Configuring the DMA for Hardware CBC Encryption

To use the DMA with hardware CBC encryption, the DMA and AES modules should be configured as follows:

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the plain text input buffer address location + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

Initialization Vector:

The initialization vector should be initialized to the HWCTRx registers.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. XOREN bits set to 01b to enable the XOR input path.
   d. EDMD set to 1 for encryption.
   e. KEYCPEN set to 1 to enable key capture at the end of the transaction.
   f. HCBCEN set to 1 to enable Hardware Cipher Block Chaining mode.
   g. The HCTREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit. When the encryption process finishes, the decryption key is available in the HWKEYx registers.

15.7.1.2. Configuring the DMA for Hardware CBC Decryption

To use the DMA with Hardware CBC decryption, the DMA and AES modules should be configured as follows:

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the cipher text input buffer address location + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the plain text output buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

Initialization Vector:

The initialization vector should be initialized to the HWCTRx registers.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. XOREN set to 10b to enable the XOR output path.
   d. EDMD set to 0 for decryption.
   e. KEYCPEN set to 0 to disable key capture at the end of the transaction.
   f. HCBCEN set to 1 to enable Hardware Cipher Block Chaining mode.
   g. The HCTREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit.

15.7.2. DMA XOR Channel Cipher Block Chaining

The block diagram of the AES module performing this encryption algorithm is shown in Figure 15.8. The block diagram of the AES module performing this decryption algorithm is shown in Figure 15.9. The active data paths in this mode are shown in red.

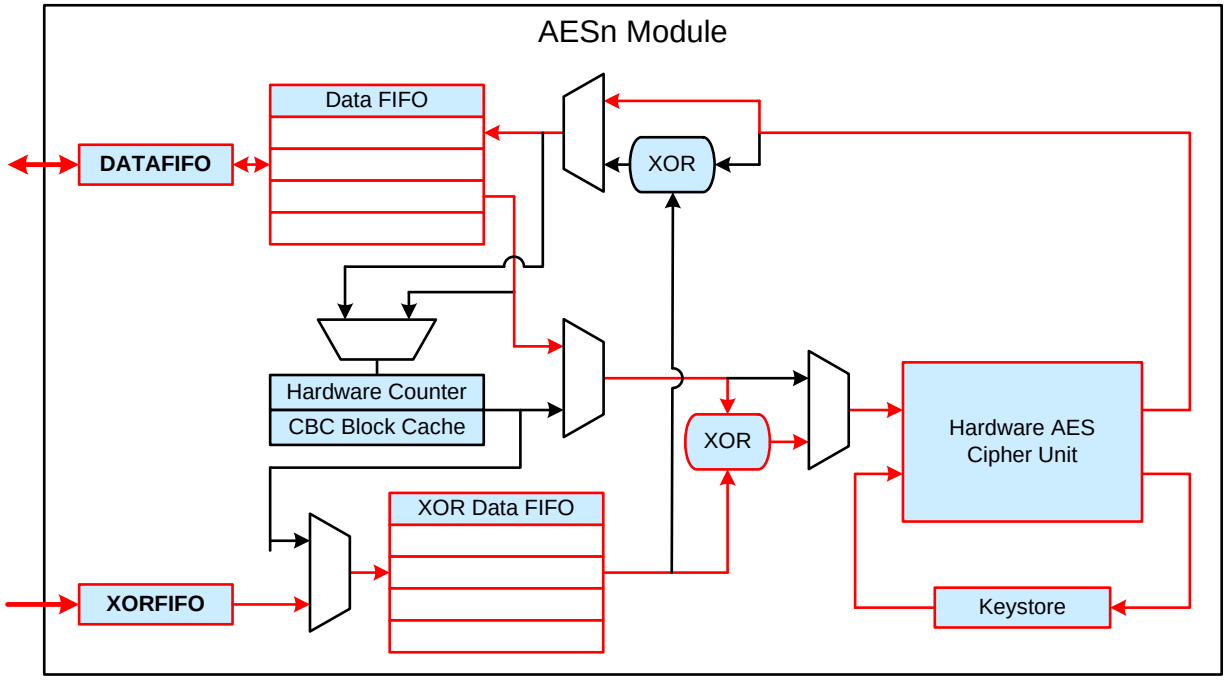


Figure 15.8. Cipher Block Chaining (CBC) AES Module Block Diagram—Encryption

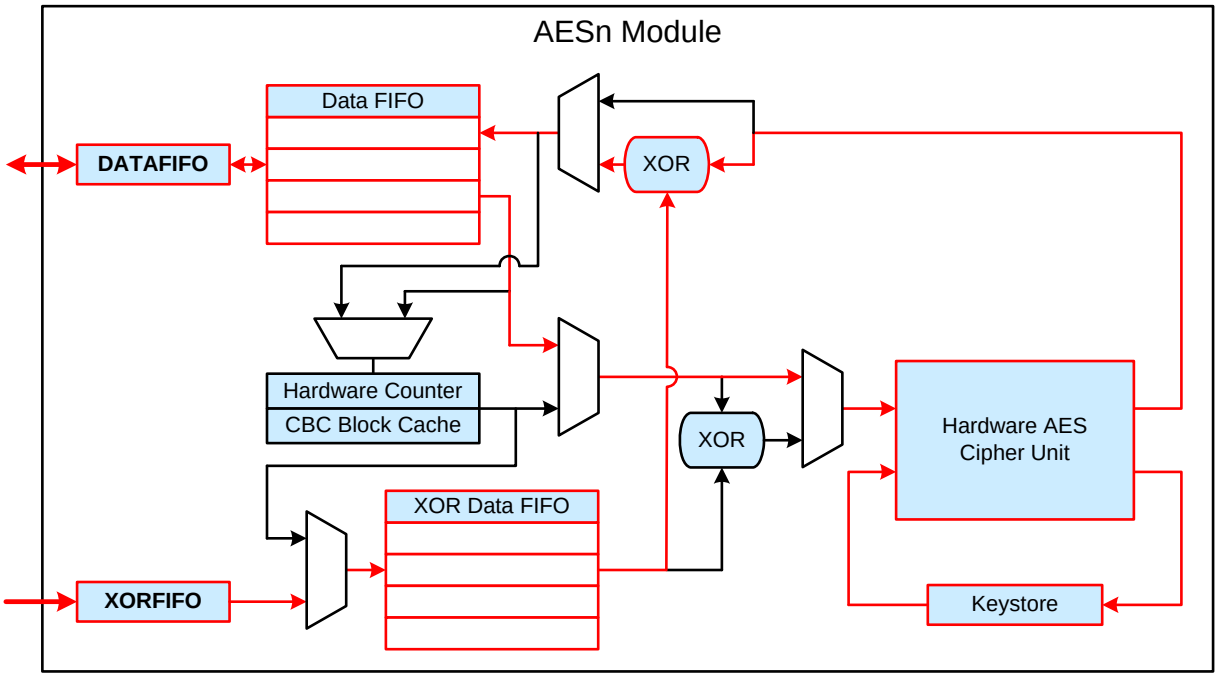


Figure 15.9. Cipher Block Chaining (CBC) AES Module Block Diagram—Decryption

15.7.2.1. Configuring the DMA for CBC Encryption

To use the DMA with CBC encryption, the DMA and AES modules should be configured as follows:

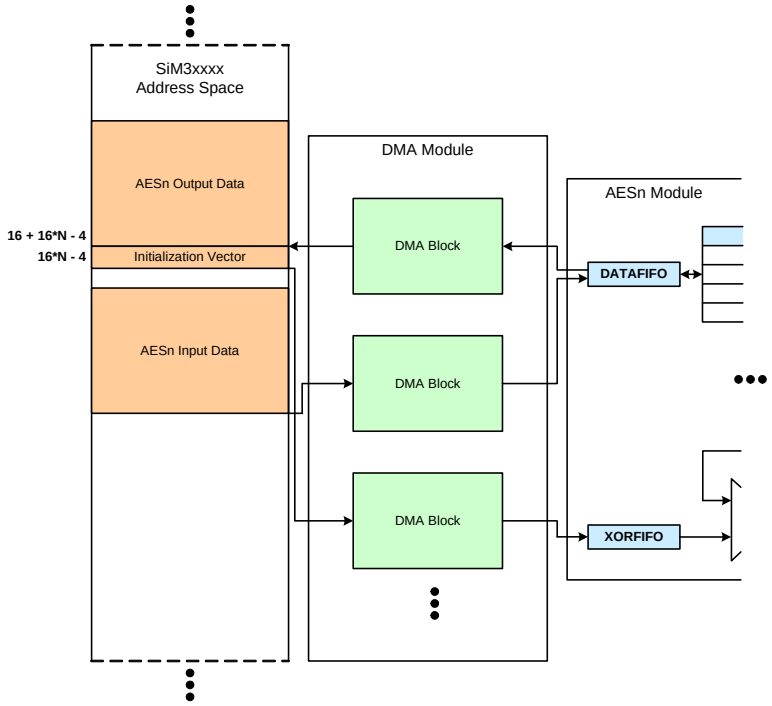


Figure 15.10. DMA XOR Channel Cipher Block Chaining Memory Setup

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the plain text input buffer address location + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

DMA XOR Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 x N – 4, where N is the number of blocks. By programming the output 16 bytes ahead of the XOR channel, the XOR channel is always one block behind (CT(n-1)).
2. Source end pointer set to the DATAFIFO register.

Initialization Vector:

The initialization vector should be initialized to the cipher text output buffer address location + 16 x N – 4.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. XOREN set to 01b to enable the XOR input path.

d. EDMD set to 1 for encryption.

e. KEYCPEN set to 1 to enable key capture at the end of the transaction.

f. The HCBCEN, HCTREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit. When the encryption process finishes, the decryption key is available in the HWKEYx registers.

15.7.2.2. Configuring the DMA for CBC Decryption

The decryption process is similar to the encryption process, except now the CT(n-1) value is taken before the data is passed through the AES module instead of after, as shown in Figure 15.5. To use the DMA with CBC decryption, the DMA and AES modules should be configured as follows:

DMA Input Channel:

1. Destination end pointer set to the DATAFIFO register.
2. Source end pointer set to the cipher text input buffer address location + 16 + 16 x N – 4, where N is the number of blocks.

DMA Output Channel:

1. Destination end pointer set to the plain text output buffer address location + 16*N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

DMA XOR Channel:

1. Destination end pointer set to the cipher text input buffer address location + 16 x N – 4, where N is the number of blocks. By programming the DMA input 16 bytes ahead of the XOR channel, the XOR channel is always one block behind (CT(n-1)).
2. Source end pointer set to the DATAFIFO register.

Initialization Vector:

The initialization vector should be initialized to the cipher text input buffer address location + 16*N – 4.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. XOREN set to 10b to enable the XOR output path.
   d. EDMD set to 0 for decryption.
   e. KEYCPEN set to 0 to disable key capture at the end of the transaction.
   f. The HCBCEN, HCTREN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit.

15.8. Using the AES Module for Counter (CTR)

The counter (CTR) cipher algorithm is a stream cipher mode which improves upon the basic ECB algorithm by adding a third block variable (a counter block in this case). This algorithm is shown in Figure 15.11.

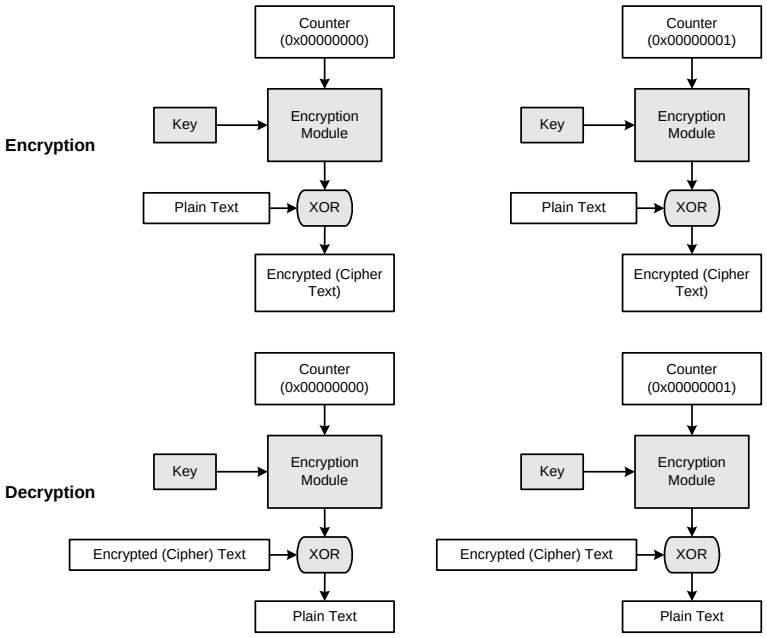


Figure 15.11. Counter (CTR) Algorithm Diagram

Similar to CBC mode, the CTR algorithm requires an initialization vector to encrypt the first block. Unlike CBC, this value is a counter instead of the previous block's output. This counter is implemented in hardware in the AES module.

The block diagram of the AES module performing this algorithm (encryption and decryption) is shown in Figure 15.12. The active data paths in this mode are shown in red.

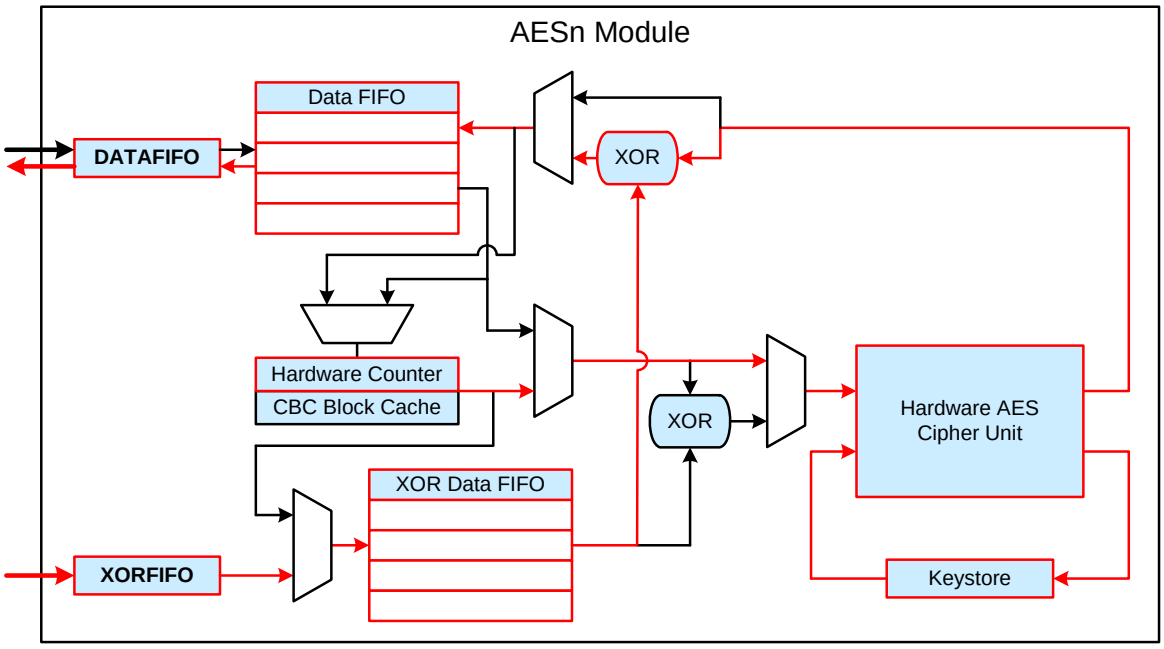


Figure 15.12. Counter (CTR) AES Module Block Diagram—Encryption and Decryption

15.8.1. Configuring the DMA for CTR Encryption

To use the DMA with CTR encryption, the DMA and AES modules should be configured as follows:

DMA Output Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

DMA XOR Channel:

1. Destination end pointer set to the plain text input buffer address location + 16 x N – 4.
2. Source end pointer set to the XORFIFO register.

Initialization Vector:

The initialization vector should be initialized to the HWCTRx registers.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. EDMD set to 1 for encryption.
   d. KEYCPEN set to 0 to disable key capture at the end of the transaction.
   e. HCTREN set to 1 to enable Hardware Counter mode.
   f. XOREN set to 10b to enable the XOR output path.
   g. The HCBCEN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit.

15.8.2. Configuring the DMA for CTR Decryption

This algorithm does not need the key capture output from the encryption process since the encryption and decryption algorithms are exactly the same.

To use the DMA with CTR decryption, the DMA and AES modules should be configured as follows:

DMA Output Channel:

1. Destination end pointer set to the plain text input buffer address location + 16 x N – 4, where N is the number of blocks.
2. Source end pointer set to the DATAFIFO register.

DMA XOR Channel:

1. Destination end pointer set to the cipher text output buffer address location + 16 x N – 4.
2. Source end pointer set to the XORFIFO register.

Initialization Vector:

The initialization vector should be initialized to the HWCTRx registers.

AES Module:

1. The XFRSIZE register should be set to N-1, where N is the number of 4-word blocks.
2. The HWKEYx registers should be written with the desired key in little endian format.
3. The CONTROL register should be set as follows:
   a. ERRIEN set to 1.
   b. KEYSIZE set to the appropriate number of bits for the key.
   c. EDMD set to 1 for encryption.
   d. KEYCPEN set to 0 to disable key capture at the end of the transaction.
   e. HCTREN set to 1 to enable Hardware Counter mode.
   f. XOREN set to 10b to enable the XOR output path.
   g. The HCBCEN, BEN, SWMDEN bits should all be cleared to 0.

Once the DMA and AES settings have been set, the transfer should be started by writing 1 to the XFRSTA bit.

15.9. Performing “In-Place” Ciphers

For the cipher examples described in Section 15.6, Section 15.7, and Section 15.8, the DMA channels are described using plain text and cipher text address offsets. However, these addresses can be the same instead of separate entities to reduce general-purpose memory usage. These in-place ciphers overwrite the plain text input with the cipher text output data.

For example, the hardware cipher block chaining algorithm executed in place is shown in Figure 15.13.

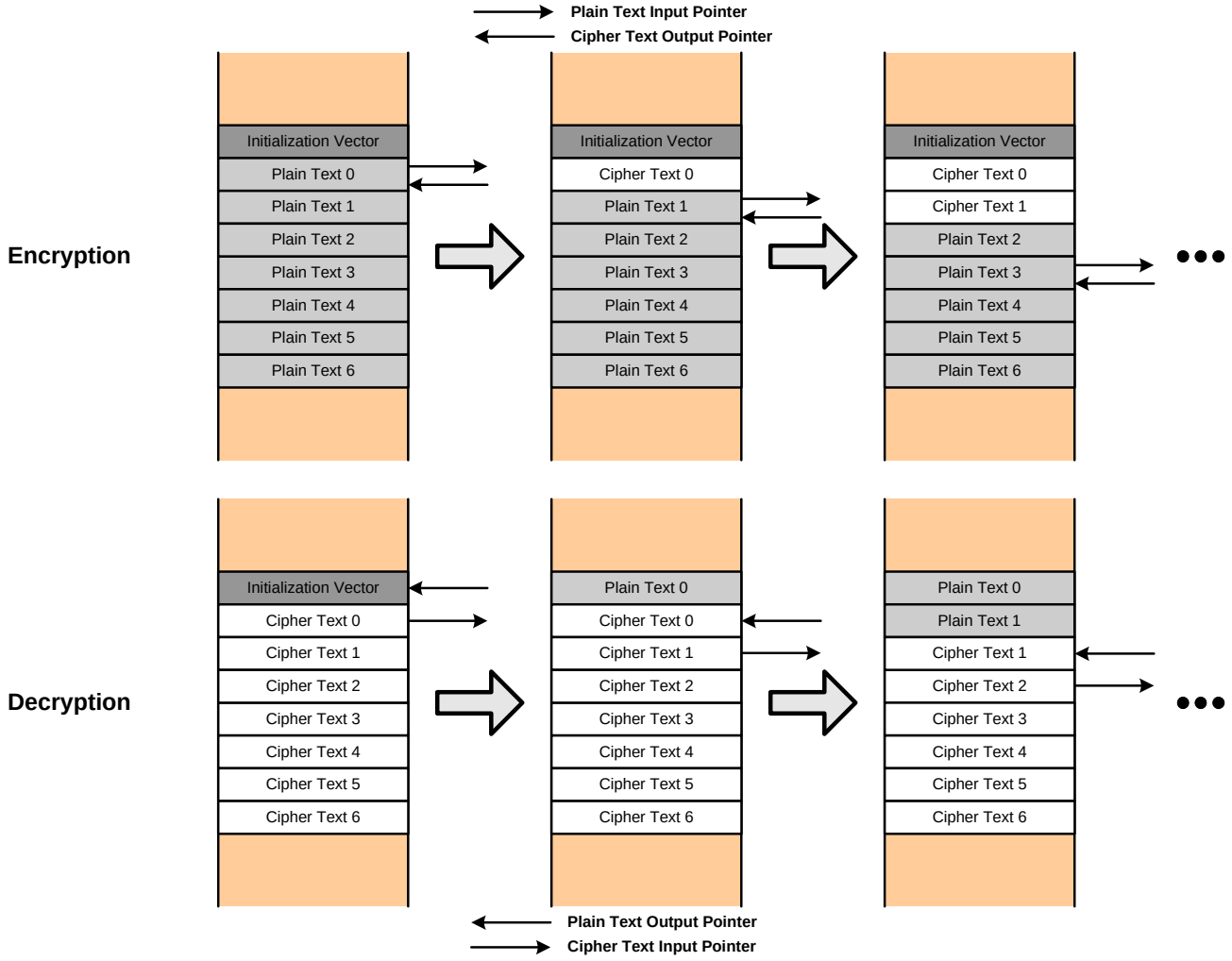


Figure 15.13. Memory Map of Hardware CBC Algorithm Performed In Place

15.10. Using the AES Module in Software Mode

Software mode (SWMDEN = 1) allows the firmware to perform smaller or more custom operations with the AES module. When software mode is enabled, the AES module will not generate any DMA requests.

In software mode, each operation must consist of 4 words and follows this general encryption or decryption sequence:

1. The RESET bit, if set, must be cleared to access the AES registers.
2. Configure the operation, including setting SWMDEN to 1.
3. Load the input/output data FIFO (DATAFIFO) with four words using word, half-word, or byte writes.
4. If XOREN is set to 01b or 10b, load the XOR data FIFO (XORFIFO) with four words using word, half-word, or byte writes.
5. Set KEYCPEN to 1 if key capture is required (EDMD must also be set to 1 for the key capture to occur).
6. Enable the operation complete interrupt by setting OCIEN to 1. Alternatively, firmware can poll XFRSTA or BUSYF.
7. Set XFRSTA to 1 to start the AES operation on the 4-word block.
8. Wait for the operation completion interrupt (or poll XFRSTA or BUSYF until the operation completes).
9. Read four words from the input/output data FIFO (DATAFIFO) with word, half-word, or byte reads to obtain the resulting cipher text output.

If key capture (KEYCPEN set to 1) was enabled for an encryption operation, then the key is overwritten. The key must be re-written if a subsequent operation is also an encryption.

The hardware counter and hardware cipher block chaining modes can be used in conjunction with software mode, but bypass mode (BEN) is not available.

15.10.1. Software Mode Error Conditions

Firmware can check the number of bytes present in each of the FIFOs using the DFIFOLVL and XFIFOLVL fields in the STATUS register.

Care must be taken when reading or writing the input/output or XOR FIFOs:

- Loading more than 16 bytes into the input/output data FIFO results in a data overrun error (DORI = 1).
- Loading more than 16 bytes into the XOR data FIFO results in a XOR data overrun error (XORI = 1).
- Attempting to read more than 16 bytes from the input/output data FIFO results in a data underrun error (DURI = 1)
- Loading less than 16 bytes into the data FIFOs prevents an operation from starting when the XFRSTA bit is set to 1.

Failure to read the data from the input/output data FIFO leaves the FIFO full, so any subsequent writes to this FIFO with new input data causes a data overrun event. Any unwanted data in the data or XOR FIFOs may be discarded by soft resetting the AES module (RESET = 1).

15.11. AES0 Registers

This section contains the detailed register descriptions for AES0 registers.

Register 15.1. AES0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESET	DBGMD	Reserved				OCIEN	ERRIEN	Reserved						KEYSIZE	
Type	RW	RW	R				RW	RW	R						RW	
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved		HCBCEN	HCTREN	XOREN		BEN	SWMDEN	Reserved					EDMD	KEYCPEN	XFRSTA
Type	R		RW	RW	RW		RW	RW	R					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_CONTROL = 0x4002_7000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 15.1. AES0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	RESET	**Module Soft Reset.** Must be cleared to access any other AES module bit. 0: AES module is not in soft reset. 1: AES module is in soft reset and none of the module bits can be accessed.
30	DBGMD	**AES Debug Mode.** 0: A debug breakpoint will cause the AES module to halt. 1: The AES module will continue to operate while the core is halted in debug mode.
29:26	Reserved	Must write reset value.
25	OCIEN	**Operation Complete Interrupt Enable.** Enables the completion interrupt for each 4-word block. 0: Disable the operation complete interrupt. 1: Enable the operation complete interrupt. An interrupt is generated when the Operation Complete Interrupt (OCI) flag is set.

Table 15.1. AES0_CONTROL Register Bit Descriptions

Bit	Name	Function
24	ERRIEN	**Error Interrupt Enable.** 0: Disable the error interrupt. 1: Enable the error interrupt. An interrupt is generated when the Input/Output Data FIFO Overun (DORI), Input/Output Data FIFO Underun (DURI), or XOR Data FIFO Overrun (XORI) flags are set.
23:18	Reserved	Must write reset value.
17:16	KEYSIZE	**Keystore Size Select.** Selects the size of the key used in the AES encryption or decryption process. 00: Key is composed of 128 bits. 01: Key is composed of 192 bits. 10: Key is composed of 256 bits. 11: Reserved.
15:14	Reserved	Must write reset value.
13	HCBCEN	**Hardware Cipher-Block Chaining Mode Enable.** Enables the Hardware Cipher-Clock Chaining (CBC) mode. This causes the XOR path to be fed automatically from hardware with CT(n-1), so there is no need for firmware or the DMA to feed the XOR path in this mode. 0: Disable hardware cipher-block chaining (CBC) mode. 1: Enable hardware cipher-block chaining (CBC) mode.
12	HCTREN	**Hardware Counter Mode Enable.** Enables the Hardware Counter Mode. 0: Disable hardware counter mode. 1: Enable hardware counter mode.
11:10	XOREN	**XOR Enable.** Enables the input or output XOR path. 00: Disable the XOR paths. 01: Enable the XOR input path, disable the XOR output path. 10: Disable the XOR input path, enable the XOR output path. 11: Reserved.
9	BEN	**Bypass AES Operation Enable.** If this bit is set to 1, the AES module hardware is bypassed, which allows firmware to use the module as a memory copy. The XOR paths (output and input) are not available in this mode. 0: Do not bypass AES operations. 1: Bypass AES operations.

Table 15.1. AES0_CONTROL Register Bit Descriptions

Bit	Name	Function
8	SWMDEN	**Software Mode Enable.** Setting this bit to 1 stops DMA requests from being generated. When this bit is 1, firmware is responsible for loading the input FIFO with 4 words, loading the XOR FIFO with 4 words (if applicable), and reading 4 words from the output FIFO after receiving the done interrupt. If this bit is 1, the transfer size register (XFRSIZE) should be cleared to 0. Bypass mode is not supported in conjunction with software mode. 0: Disable software mode. 1: Enable software mode.
7:3	Reserved	Must write reset value.
2	EDMD	**Encryption/Decryption Mode.** 0: AES module performs a decryption operation 1: AES module performs an encryption operation.
1	KEYCPEN	**Key Capture Enable.** If this bit is set to 1, the current key in the keystore is overwritten during the module operation. In the case where EDMD is set to 1 (encryption operation), this generated key is the proper decryption key after the last block is complete. If SWMDEN is cleared to 0, the hardware only overwrites the key on the last block of the DMA transfer. If SWMDEN is set to 1, the hardware overwrites the key for each operation KEYCPEN is set to 1. 0: Disable key capture. 1: Enable key capture.
0	XFRSTA	**AES Transfer Start.** If SWMDEN is set to 1, setting this bit to 1 starts an AES module operation on the 4-word block. This bit is automatically cleared when the 4-word operation completes. If SWMDEN is cleared to 0, setting this bit to 1 starts a series of AES module operations until the XFRSIZE register counts down to 0. This bit is automatically cleared when the XFRSIZE regiser is 0 and the current operation completes.

Register 15.2. AES0_XFRSIZE: Number of Blocks

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					XFRSIZE										
Type	R					RW										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_XFRSIZE = 0x4002_7010

Table 15.2. AES0_XFRSIZE Register Bit Descriptions

Bit	Name	Function
31:11	Reserved	Must write reset value.
10:0	XFRSIZE	**Transfer Size.** The number of 4-word blocks such that XFRSIZE + 1 blocks will be processed and transferred by the AES module. This value must match the DMA transfer size for each relevant channel. This value is automatically decremented by hardware as each block operation completes. When the SWMDEN bit is set to 1, this register should be set to 0.

Register 15.3. AES0_DATAFIFO: Input/Output Data FIFO Access

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATAFIFO[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATAFIFO[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

AES0_DATAFIFO = 0x4002_7020

Table 15.3. AES0_DATAFIFO Register Bit Descriptions

Bit	Name	Function
31:0	DATAFIFO	**Input/Output Data FIFO Access.** This data register interfaces with the AES Input/Output data FIFO. Reads from this register will result in data pops from the Input/Output FIFO. Writes to this register will result in data pushes to the Input/Output FIFO. Input/Output data FIFO overflows and underruns will result in an error interrupt (if enabled). Reads and writes may be word, half-word, or byte length and should always be right-justified. Byte-wide reads and writes should always access DATAFIFO[7:0], half-word reads and writes should always access DATAFIFO[15:0], and word reads and writes should always access DATAFIFO[31:0]. For DMA operations, this register should be targeted by the DMA input data and DMA output data in non-incrementing mode. The DMA RPOWER bits must be set as follows: For byte-accesses: RPOWER can be 0,1,2,3 or 4 For half-word accesses: RPOWER can be 0,1,2 or 3 For word accesses: RPOWER can be 0,1 or 2 For software operations (SWMDEN bit is set to 1), the DATAFIFO register must be written until the FIFO is full, or read until FIFO is empty, before the AES operation can be initiated using the XFRSTA bit.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

Register 15.4. AES0_XORFIFO: XOR Data FIFO Access

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	XORFIFO[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	XORFIFO[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

AES0_XORFIFO = 0x4002_7030

Table 15.4. AES0_XORFIFO Register Bit Descriptions

Bit	Name	Function
31:0	XORFIFO	**XOR Data FIFO Access.** This data register interfaces with the AES XOR data FIFO. Reads from this register have no effect. Writes to this register will result in data pushes to the XOR data FIFO. XOR data FIFO overflows will result in an error interrupt (if enabled). Writes may be word, half-word, or byte length, and should always be right-justified. Byte-wide writes should always access XORFIFO[7:0], half-word writes should always access XORFIFO[15:0], and word writes should always access DATAFIFO[31:0]. For DMA operations, this register should be targeted by the DMA input XOR data and DMA output data in non-incrementing mode. The DMA RPOWER bits must be set as follows: For byte-accesses: RPOWER can be 0,1,2,3 or 4 For half-word accesses: RPOWER can be 0,1,2 or 3 For word accesses: RPOWER can be 0,1 or 2 For software operations (SWMDEN bit is set to 1), the XORFIFO register must be written until the FIFO is full before the AES operation can be initiated using the XFRSTA bit.

Register 15.5. AES0_HWKEY0: Hardware Key Word 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY0 = 0x4002_7040

Table 15.5. AES0_HWKEY0 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY0	**Hardware Key Word 0.** This register contains word 0 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.6. AES0_HWKEY1: Hardware Key Word 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY1 = 0x4002_7050

Table 15.6. AES0_HWKEY1 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY1	**Hardware Key Word 1.** This register contains word 1 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.7. AES0_HWKEY2: Hardware Key Word 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY2 = 0x4002_7060

Table 15.7. AES0_HWKEY2 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY2	**Hardware Key Word 2.** This register contains word 2 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.8. AES0_HWKEY3: Hardware Key Word 3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY3[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY3[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY3 = 0x4002_7070

Table 15.8. AES0_HWKEY3 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY3	**Hardware Key Word 3.** This register contains word 3 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.9. AES0_HWKEY4: Hardware Key Word 4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY4[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY4[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY4 = 0x4002_7080

Table 15.9. AES0_HWKEY4 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY4	**Hardware Key Word 4.** This register contains word 4 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.10. AES0_HWKEY5: Hardware Key Word 5

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY5[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY5[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY5 = 0x4002_7090

Table 15.10. AES0_HWKEY5 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY5	**Hardware Key Word 5.** This register contains word 5 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.11. AES0_HWKEY6: Hardware Key Word 6

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY6[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY6[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY6 = 0x4002_70A0

Table 15.11. AES0_HWKEY6 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY6	**Hardware Key Word 6.** This register contains word 6 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.12. AES0_HWKEY7: Hardware Key Word 7

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWKEY7[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWKEY7[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWKEY7 = 0x4002_70B0

Table 15.12. AES0_HWKEY7 Register Bit Descriptions

Bit	Name	Function
31:0	HWKEY7	**Hardware Key Word 7.** This register contains word 7 of the keystore. If the KEYCPEN bit is set to 1, the new key will be accessible using these registers after the old key is overwritten.

Register 15.13. AES0_HWCTR0: Hardware Counter Word 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWCTR0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWCTR0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWCTR0 = 0x4002_70C0

Table 15.13. AES0_HWCTR0 Register Bit Descriptions

Bit	Name	Function
31:0	HWCTR0	**Hardware Counter Word 0.** This register contains word 0 of the 128-bit hardware counter. These registers are little endian, and HWCTR0 holds the least-significant word of the counter. When the HCTREN bit is set to 1, this register should be written with the initial counter value to seed the encryption or decryption process for a block of operations. Reading this register always reflects the current value of the hardware counter. Firmware should not modify the contents of this register when using hardware CBC mode (HCBCEN = 1).

Register 15.14. AES0_HWCTR1: Hardware Counter Word 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWCTR1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWCTR1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWCTR1 = 0x4002_70D0

Table 15.14. AES0_HWCTR1 Register Bit Descriptions

Bit	Name	Function
31:0	HWCTR1	**Hardware Counter Word 1.** This register contains word 1 of the 128-bit hardware counter. These registers are little endian. When the HCTREN bit is set to 1, this register should be written with the initial counter value to seed the encryption or decryption process for a block of operations. Reading this register always reflects the current value of the hardware counter. Firmware should not modify the contents of this register when using hardware CBC mode (HCBCEN = 1).

Register 15.15. AES0_HWCTR2: Hardware Counter Word 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWCTR2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWCTR2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWCTR2 = 0x4002_70E0

Table 15.15. AES0_HWCTR2 Register Bit Descriptions

Bit	Name	Function
31:0	HWCTR2	**Hardware Counter Word 2.** This register contains word 2 of the 128-bit hardware counter. These registers are little endian. When the HCTREN bit is set to 1, this register should be written with the initial counter value to seed the encryption or decryption process for a block of operations. Reading this register always reflects the current value of the hardware counter. Firmware should not modify the contents of this register when using hardware CBC mode (HCBCEN = 1).

Register 15.16. AES0_HWCTR3: Hardware Counter Word 3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HWCTR3[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	HWCTR3[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_HWCTR3 = 0x4002_70F0

Table 15.16. AES0_HWCTR3 Register Bit Descriptions

Bit	Name	Function
31:0	HWCTR3	**Hardware Counter Word 3.** This register contains word 3 of the 128-bit hardware counter. These registers are little endian. When the HCTREN bit is set to 1, this register should be written with the initial counter value to seed the encryption or decryption process for a block of operations. Reading this register always reflects the current value of the hardware counter. Firmware should not modify the contents of this register when using hardware CBC mode (HCBCEN = 1).

Register 15.17. AES0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OCI	XORI	DORI	DURI	Reserved			BUSYF	Reserved							
Type	RW	RW	RW	RW	R			R	R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			XFIFOLVL					Reserved			DFIFOLVL				
Type	R			R					R			R				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

AES0_STATUS = 0x4002_7100

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 15.17. AES0_STATUS Register Bit Descriptions

Bit	Name	Function
31	OCI	**Operation Complete Interrupt Flag.** This bit is set to 1 by hardware when the current AES operation is complete. This bit must be cleared by firmware.
30	XORI	**XOR Data FIFO Overrun Interrupt Flag.** This bit is set to 1 by hardware when an XOR data FIFO overrun occurs. This flag must be cleared by firmware.
29	DORI	**Input/Output Data FIFO Overrun Interrupt Flag.** This bit is set to 1 by hardware when an input/output data FIFO overrun occurs. This flag must be cleared by firmware.
28	DURI	**Input/Output Data FIFO Underrun Interrupt Flag.** This bit is set to 1 by hardware when an input/output data FIFO underrun occurs. This flag must be cleared by firmware.
27:25	Reserved	Must write reset value.
24	BUSYF	**Module Busy Flag.** 0: AES module is not busy. 1: AES module is completing an operation.
23:13	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 15.17. AES0_STATUS Register Bit Descriptions

Bit	Name	Function
12:8	XFIFOLVL	**XOR Data FIFO Level.** 00000: XOR data FIFO is empty. 00001: XOR data FIFO contains 1 byte. 00010: XOR data FIFO contains 2 bytes. 00011: XOR data FIFO contains 3 bytes. 00100: XOR data FIFO contains 4 bytes. 00101: XOR data FIFO contains 5 bytes. 00110: XOR data FIFO contains 6 bytes. 00111: XOR data FIFO contains 7 bytes. 01000: XOR data FIFO contains 8 bytes. 01001: XOR data FIFO contains 9 bytes. 01010: XOR data FIFO contains 10 bytes. 01011: XOR data FIFO contains 11 bytes. 01100: XOR data FIFO contains 12 bytes. 01101: XOR data FIFO contains 13 bytes. 01110: XOR data FIFO contains 14 bytes. 01111: XOR data FIFO contains 15 bytes. 10000: XOR data FIFO contains 16 bytes (full). 10001-11111: Reserved.
7:5	Reserved	Must write reset value.
4:0	DFIFOLVL	**Input/Output Data FIFO Level.** 00000: Input/Output data FIFO is empty. 00001: Input/Output data FIFO contains 1 byte. 00010: Input/Output data FIFO contains 2 bytes. 00011: Input/Output data FIFO contains 3 bytes. 00100: Input/Output data FIFO contains 4 bytes. 00101: Input/Output data FIFO contains 5 bytes. 00110: Input/Output data FIFO contains 6 bytes. 00111: Input/Output data FIFO contains 7 bytes. 01000: Input/Output data FIFO contains 8 bytes. 01001: Input/Output data FIFO contains 9 bytes. 01010: Input/Output data FIFO contains 10 bytes. 01011: Input/Output data FIFO contains 11 bytes. 01100: Input/Output data FIFO contains 12 bytes. 01101: Input/Output data FIFO contains 13 bytes. 01110: Input/Output data FIFO contains 14 bytes. 01111: Input/Output data FIFO contains 15 bytes. 10000: Input/Output data FIFO contains 16 bytes (full). 10001-11111: Reserved.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

15.12. AES0 Register Memory Map

Table 15.18. AES0 Memory Map

Register Name	AES0_CONTROL	AES0_XFRSIZE	AES0_DATAFIFO	AES0_XORFIFO	AES0_HWKEY0
ALL Address	0x4002_7000	0x4002_7010	0x4002_7020	0x4002_7030	0x4002_7040
Access Methods	ALL | SET | CLR	ALL	ALL	ALL	ALL
Bit 31	RESET	Reserved	DATAFIFO	XORFIFO	HWKEY0
Bit 30	DBGMD				
Bit 29	Reserved				
Bit 28					
Bit 27					
Bit 26					
Bit 25	OCIEN				
Bit 24	ERRIEN				
Bit 23	Reserved				
Bit 22					
Bit 21					
Bit 20					
Bit 19					
Bit 18					
Bit 17	KEYSIZE				
Bit 16					
Bit 15	Reserved				
Bit 14					
Bit 13	HCBCEN				
Bit 12	HCTREN				
Bit 11	XOREN	XFRSIZE			
Bit 10					
Bit 9	BEN				
Bit 8	SWMDEN				
Bit 7	Reserved				
Bit 6					
Bit 5					
Bit 4					
Bit 3					
Bit 2	EDMD				
Bit 1	KEYCPEN				
Bit 0	XFRSTA				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 15.18. AES0 Memory Map

Register Name	AES0_HWKEY1	AES0_HWKEY2	AES0_HWKEY3	AES0_HWKEY4	AES0_HWKEY5
ALL Address	0x4002_7050	0x4002_7060	0x4002_7070	0x4002_7080	0x4002_7090
Access Methods	ALL	ALL	ALL	ALL	ALL
Bit 31	HWKEY1	HWKEY2	HWKEY3	HWKEY4	HWKEY5
Bit 30					
Bit 29					
Bit 28					
Bit 27					
Bit 26					
Bit 25					
Bit 24					
Bit 23					
Bit 22					
Bit 21					
Bit 20					
Bit 19					
Bit 18					
Bit 17					
Bit 16					
Bit 15					
Bit 14					
Bit 13					
Bit 12					
Bit 11					
Bit 10					
Bit 9					
Bit 8					
Bit 7					
Bit 6					
Bit 5					
Bit 4					
Bit 3					
Bit 2					
Bit 1					
Bit 0					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 15.18. AES0 Memory Map

Register Name	AES0_HWKEY6	AES0_HWKEY7	AES0_HWCTR0	AES0_HWCTR1	AES0_HWCTR2
ALL Address	0x4002_70A0	0x4002_70B0	0x4002_70C0	0x4002_70D0	0x4002_70E0
Access Methods	ALL	ALL	ALL	ALL	ALL
Bit 31	HWKEY6	HWKEY7	HWCTR0	HWCTR1	HWCTR2
Bit 30					
Bit 29					
Bit 28					
Bit 27					
Bit 26					
Bit 25					
Bit 24					
Bit 23					
Bit 22					
Bit 21					
Bit 20					
Bit 19					
Bit 18					
Bit 17					
Bit 16					
Bit 15					
Bit 14					
Bit 13					
Bit 12					
Bit 11					
Bit 10					
Bit 9					
Bit 8					
Bit 7					
Bit 6					
Bit 5					
Bit 4					
Bit 3					
Bit 2					
Bit 1					
Bit 0					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 15.18. AES0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
AES0_HWCTR3	0x4002_70F0	ALL	HWCTR3																															
AES0_STATUS	0x4002_7100	ALL | SET | CLR	OCI	XORI	DORI	DURI	Reserved			BUSYF	Reserved											XFIFOLVL				Reserved			DFIFOLVL					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

16. Comparator (CMP0 and CMP1)

This section describes the Comparator (CMP) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the CMP block, which is used by CMP0 and CMP1 on all device families covered in this document.

16.1. Comparator Features

The comparator takes two analog input voltages and outputs the relationship between these voltages (less than or greater than). The comparator module includes the following features:

- Multiple sources for the positive and negative inputs, including VBAT, VREF, and up to 8 I/O pins.
- Two outputs are available: a digital synchronous latched output and a digital asynchronous raw output.
- Programmable hysteresis and response time.
- Falling or rising edge interrupt options on the comparator output.
- Integrated 6-bit reference DAC.

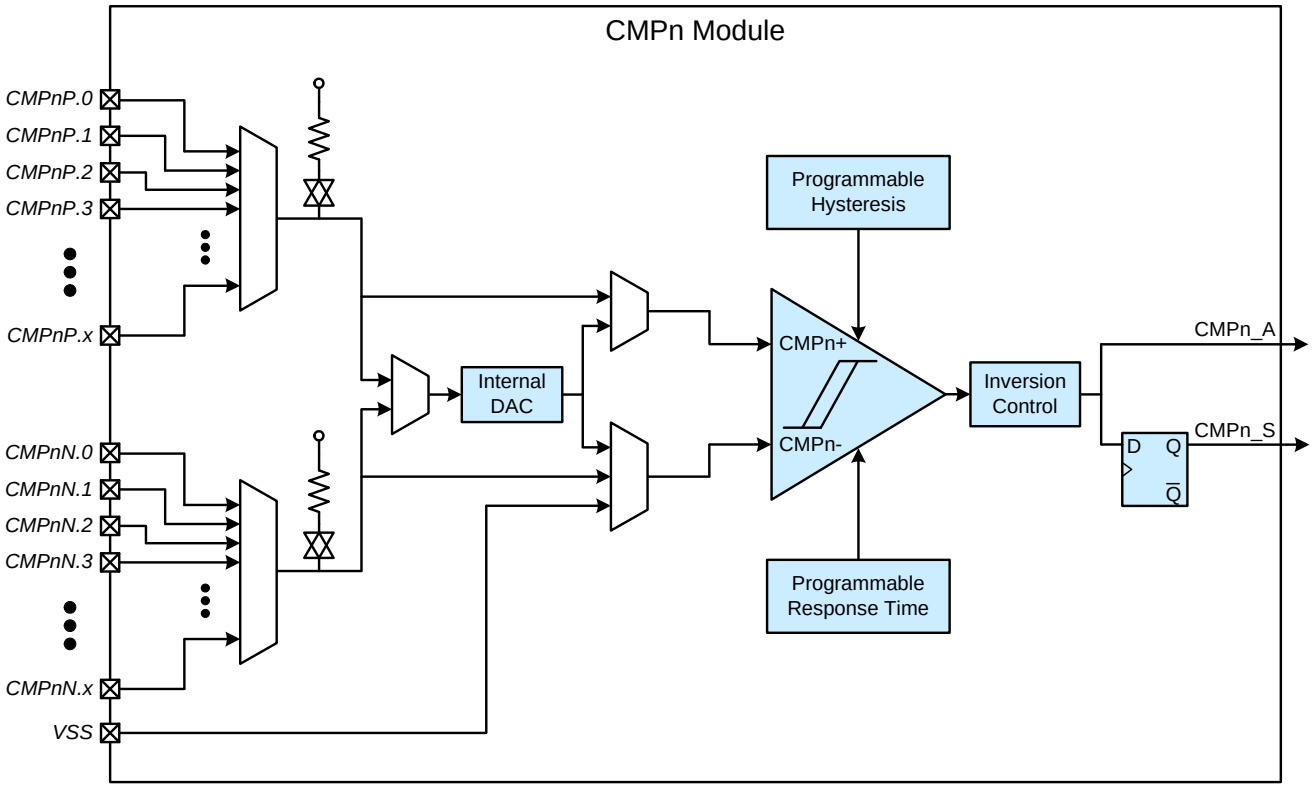


Figure 16.1. Comparator Block Diagram

16.2. Overview

The comparator offers programmable response time and hysteresis, an analog input multiplexer, and two outputs that are optionally available at the port bank pins: a digital synchronous latched output (CMPn_S) and a digital asynchronous raw output (CMPn_A). The asynchronous CMPn_A signal is available even when the system clock is not active, allowing the comparator to operate and generate an output when the device is in certain low power modes.

The comparator also features an internal DAC that may be used to create a firmware-programmable threshold voltage. The comparator DAC output level (DACLVL) field configures the DAC output voltage, and the input mux select (INMUX) field enables the DAC output to the input of the comparator.

16.3. Inputs

When enabled (CMPEN = 1), the comparator performs an analog comparison of the voltage levels at its positive (CMPn+) and negative (CMPn–) inputs. The CMPn+ and CMPn– inputs connect to select internal supplies or external port pins through analog input multiplexers, configured by the positive analog input mux select (PMUX) and negative analog input mux select (NMUX) fields in the MODE register. Any port bank pins selected as comparator inputs should be configured as analog inputs, as described in the port configuration section. The CMP0 and CMP1 input channels vary between different package options, and are shown in table Table 16.1 and Table 16.2. Note that for some selections, other device circuitry must be enabled.

The CMPn+ and CMPn– inputs have weak internal pull-ups that may be enabled by setting the positive input weak pullup enable (PWPUEN) and negative input weak pullup enable (NWPUEN) bits.

Table 16.1. CMP0 Input Channels

CMP0 Input	CMP0 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
CMP0P.0	External Positive Input	PB0.0	PB0.0	PB0.0
CMP0P.1	External Positive Input	PB0.4	PB0.3	PB0.3
CMP0P.2	External Positive Input	PB1.0	PB1.0	Reserved
CMP0P.3	External Positive Input	PB1.8	PB1.7	Reserved
CMP0P.4	External Positive Input	PB2.0	PB2.0	PB2.0
CMP0P.5	External Positive Input	PB2.4	PB2.4	PB2.4
CMP0P.6	External Positive Input	PB3.4	PB3.2	Reserved
CMP0P.7	External Positive Input	PB3.8	PB3.4	Reserved
CMP0P.8	Internal Positive Input	VREF Pin		
CMP0P.9	Internal Positive Input	Reserved		
CMP0P.10	Internal Positive Input	Temperature Sensor Output		
CMP0P.11	Internal Positive Input	Low Power Charge Pump Output		
CMP0P.12	Internal Positive Input	Digital LDO Output		

Table 16.1. CMP0 Input Channels (Continued)

CMP0 Input	CMP0 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
CMP0P.13	Internal Positive Input	Memory LDO Output		
CMP0P.14	Internal Positive Input	Analog LDO Output		
CMP0N.0	External Negative Input	PB0.1	PB0.1	PB0.1
CMP0N.1	External Negative Input	PB0.9	PB0.8	PB0.2
CMP0N.2	External Negative Input	PB1.1	PB1.1	Reserved
CMP0N.3	External Negative Input	PB1.9	PB1.8	Reserved
CMP0N.4	External Negative Input	PB2.1	Reserved	PB2.1
CMP0N.5	External Negative Input	PB2.5	PB2.5	PB2.5
CMP0N.6	External Negative Input	PB3.5	PB3.3	Reserved
CMP0N.7	External Negative Input	PB3.9	PB3.5	PB3.0
CMP0N.8	Internal Negative Input	VREF Pin		
CMP0N.9	Internal Negative Input	VBAT		
CMP0N.10	Internal Negative Input	VDC		
CMP0N.11	Internal Negative Input	Digital LDO Output		
CMP0N.12	Internal Negative Input	Memory LDO Output		
CMP0N.13	Internal Negative Input	Analog LDO Output		
CMP0N.14	Internal Negative Input	VIO		

Table 16.2. CMP1 Input Channels

CMP1 Input	CMP1 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
CMP1P.0	External Positive Input	PB0.2	PB0.1	PB0.1
CMP1P.1	External Positive Input	PB0.10	PB0.9	Reserved
CMP1P.2	External Positive Input	PB1.2	PB1.2	PB0.7
CMP1P.3	External Positive Input	PB1.10	PB1.9	Reserved

Table 16.2. CMP1 Input Channels (Continued)

CMP1 Input	CMP1 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
CMP1P.4	External Positive Input	Reserved	Reserved	PB2.2
CMP1P.5	External Positive Input	PB2.6	PB2.6	PB2.6
CMP1P.6	External Positive Input	PB3.6	Reserved	Reserved
CMP1P.7	External Positive Input	PB3.10	PB3.6	PB3.1
CMP1P.8	Internal Positive Input	VREF Pin		
CMP1P.9	Internal Positive Input	Reserved		
CMP1P.10	Internal Positive Input	Temperature Sensor Output		
CMP1P.11	Internal Positive Input	Low Power Charge Pump Output		
CMP1P.12	Internal Positive Input	Digital LDO Output		
CMP1P.13	Internal Positive Input	Memory LDO Output		
CMP1P.14	Internal Positive Input	Analog LDO Output		
CMP1N.0	External Negative Input	PB0.3	PB0.2	PB0.2
CMP1N.1	External Negative Input	PB0.11	Reserved	Reserved
CMP1N.2	External Negative Input	PB1.3	PB1.3	PB0.8
CMP1N.3	External Negative Input	PB1.11	PB1.10	Reserved
CMP1N.4	External Negative Input	Reserved	Reserved	PB2.3
CMP1N.5	External Negative Input	PB2.7	PB2.7	PB2.7
CMP1N.6	External Negative Input	PB3.7	Reserved	Reserved
CMP1N.7	External Negative Input	PB3.11	PB3.7	PB3.2
CMP1N.8	Internal Negative Input	VREF Pin		
CMP1N.9	Internal Negative Input	VBAT		
CMP1N.10	Internal Negative Input	VDC		
CMP1N.11	Internal Negative Input	Digital LDO Output		
CMP1N.12	Internal Negative Input	Memory LDO Output		

Table 16.2. CMP1 Input Channels (Continued)

CMP1 Input	CMP1 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
CMP1N.13	Internal Negative Input	Analog LDO Output		
CMP1N.14	Internal Negative Input	VIO		

The CMPm module offers several different input modes shown in Table 16.3, configurable via the input mux select (INMUX) field.

Table 16.3. Comparator Input Modes

INMUX Value	Input to CMPn+	Input to CMPn-
0	positive analog input mux output	negative analog input mux output
1	positive analog input mux output	VSS
2	DAC output with positive analog input mux connected to internal DAC reference	negative analog input mux output
3	positive analog input mux output	DAC output with negative analog input mux connected to internal DAC reference

Figure 16.2, Figure 16.3, Figure 16.4, and Figure 16.5 illustrate each of these configurations.

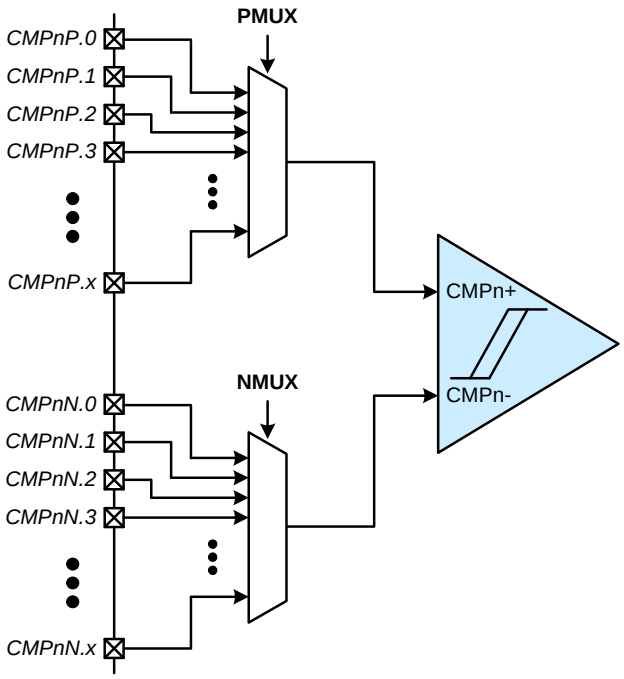


Figure 16.2. Comparator Input Mode 0 (INMUX = 0)

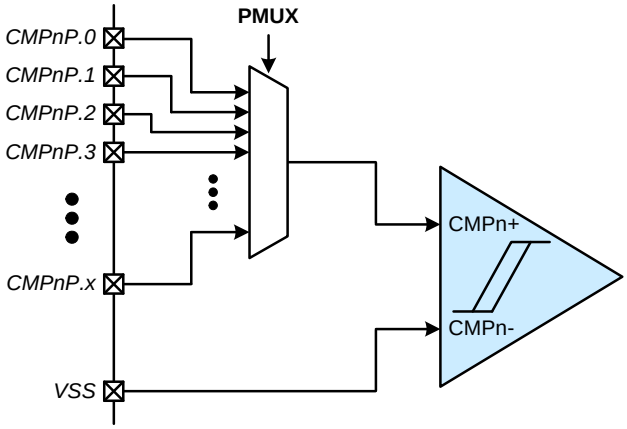


Figure 16.3. Comparator Input Mode 1 (INMUX = 1)

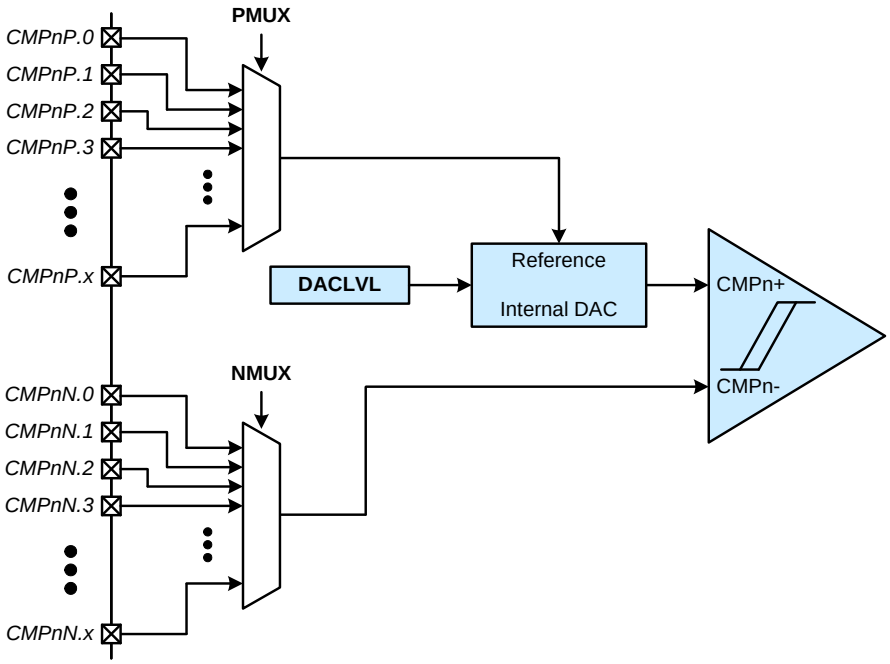


Figure 16.4. Comparator Input Mode 2 (INMUX = 2)

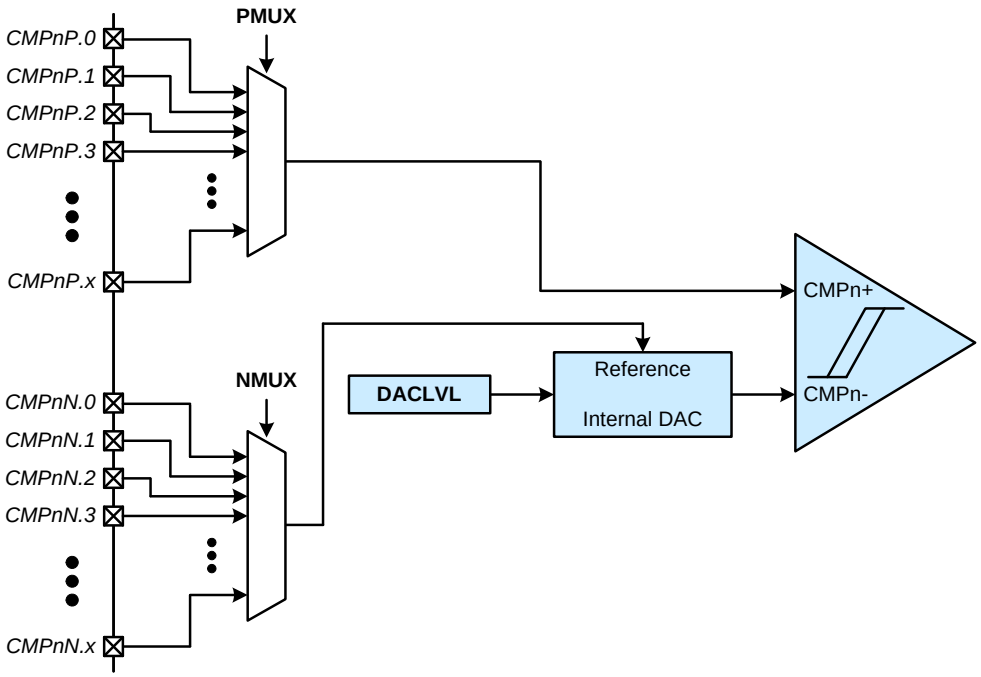


Figure 16.5. Comparator Input Mode 3 (INMUX = 3)

16.4. Outputs

When enabled (CMPEN = 1) and non-inverted (INVEN = 0), the comparator will output a 1 if the voltage at the positive input (CMPn+) is higher than the voltage at the negative input (CMPn–). Firmware can set the INVEN bit to 1 to invert the comparator output polarity.

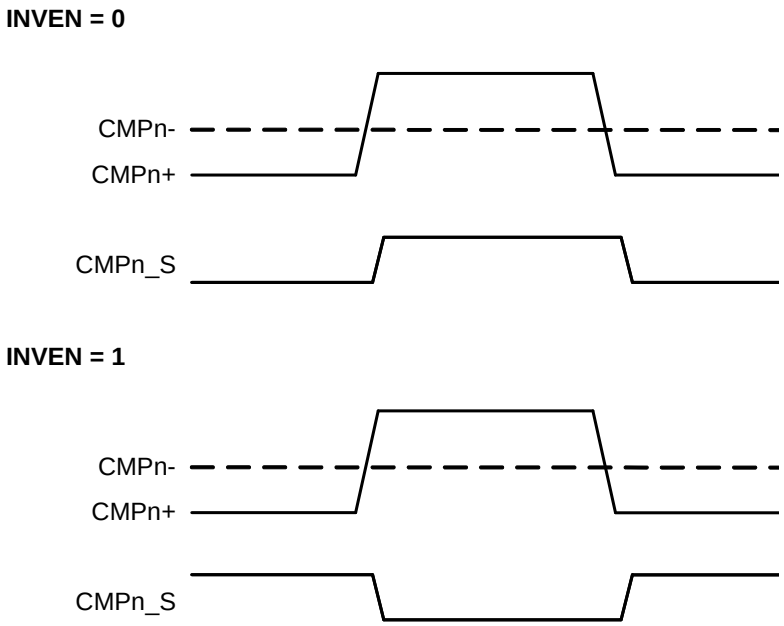


Figure 16.6. Output Configurations

The synchronous output (CMPn_S) is synchronized with the APB clock and may be polled by firmware, used as an interrupt source, or routed to a port bank pin through the crossbar. The asynchronous comparator output (CMPn_A) is not synchronized with the APB clock and can be used by low power mode wake-up logic and reset decision logic, or routed to a port bank pin through the crossbar.

When the module is disabled (CMPEN = 0), the comparator will output a static 0 (INVEN = 0) or 1 (INVEN = 1).

16.5. Response Time

The comparator response time may be configured in firmware using the CMPMD field. There are four settings available, from Mode 0 (fastest response time and highest power) to Mode 3 (slowest response time and lowest power). For lower power applications, selecting a slower response time reduces the module's active supply current.

The comparator rising edge and falling edge response times are typically not equal. The device data sheet contains comparator timing and supply current specifications.

16.6. Hysteresis

The comparator module features programmable hysteresis that can be used to stabilize the comparator output when a transition occurs on the input. The comparator output will not transition until the voltage on the comparator CMPn+ input exceeds the threshold voltage on the CMPn– input by the amount programmed in the positive hysteresis control (CMPHYP) field. Similarly, the comparator output will not transition until the voltage on the comparator CMPn+ input falls below the threshold voltage on the CMPn– input by the amount programmed in the negative hysteresis control (CMPHYN) field.

Figure 16.7 illustrates the programmable comparator hysteresis.

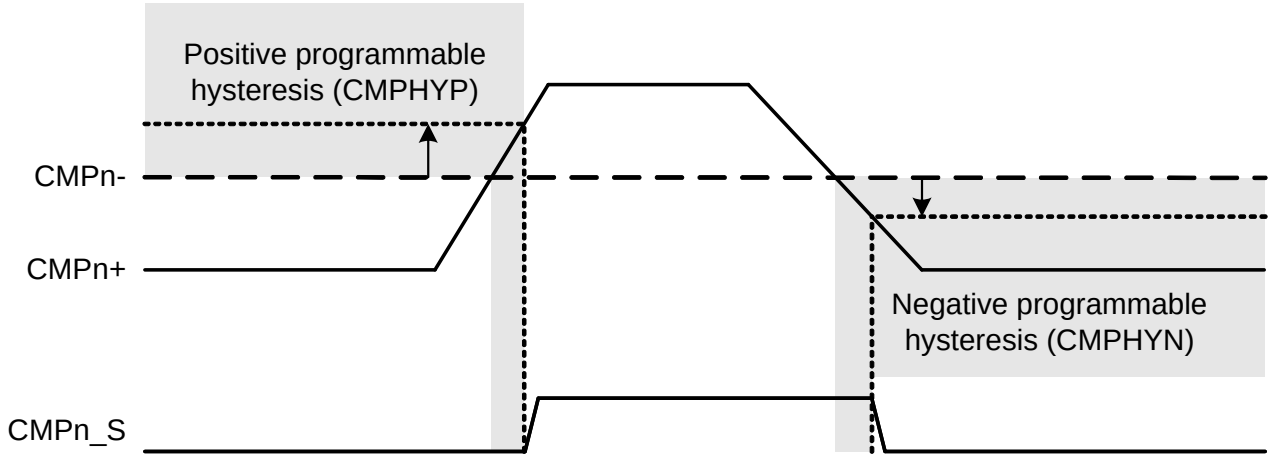


Figure 16.7. Comparator Hysteresis

Both the positive and negative hysteresis control fields may be configured for 5, 10, or 20 mV hysteresis. Firmware can disable positive or negative hysteresis by clearing CMPHYP or CMPHYN to 0.

16.7. Interrupts and Flags

The rising-edge (CMPRI) and falling-edge (CMPFI) interrupt flags allow firmware to determine whether the comparator had a rising-edge or falling-edge output transition. The rising-edge interrupt enable (RIEN) and falling-edge interrupt enable (FIEN) bits can enable these flags as an interrupt source. The module hardware sets the CMPRI and CMPFI flags when a corresponding rising or falling edge is detected, regardless of the interrupt enable state. Once set, these flags remain set until cleared by firmware.

The comparator may detect false rising and falling edges during power-on or after changes are made to the hysteresis or response time control bits. Firmware should explicitly clear the rising-edge and falling-edge flags to 0 a short time after enabling the comparator or changing the mode bits.

16.8. CMP0 and CMP1 Registers

This section contains the detailed register descriptions for CMP0 and CMP1 registers.

Register 16.1. CMPn_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMPEN	CMPOUT	Reserved													
Type	RW	R	R													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	CMPRI	CMPFI	Reserved												
Type	R	RW	RW	R												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

CMP0_CONTROL = 0x4001_F000

CMP1_CONTROL = 0x4002_0000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 16.4. CMPn_CONTROL Register Bit Descriptions

Bit	Name	Function
31	CMPEN	**Comparator Enable.** 0: Disable the comparator. 1: Enable the comparator.
30	CMPOUT	**Output State.** This bit indicates the logic level of the comparator output. When INVEN is set, it directly inverts the meaning of this bit. 0: Voltage on CMP+ < CMP- (INVEN = 0). 1: Voltage on CMP+ > CMP- (INVEN = 0).
29:15	Reserved	Must write reset value.
14	CMPRI	**Rising Edge Interrupt Flag.** 0: No comparator rising edge has occurred since this flag was last cleared. 1: A comparator rising edge occurred since last flag was cleared.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 16.4. CMPn_CONTROL Register Bit Descriptions

Bit	Name	Function
13	CMPFI	**Falling Edge Interrupt Flag.** 0: No comparator falling edge has occurred since this flag was last cleared. 1: A comparator falling edge occurred since last flag was cleared.
12:0	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 16.2. CMPn_MODE: Input and Module Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	INVEN	Reserved		CMPHYP		CMPHYN		PWPUEN	NWPUEN	DACLVL					
Type	RW	RW	R		RW		RW		RW	RW	RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	RIEN	FIEN	Reserved	CMPMD		INMUX		PMUX				NMUX			
Type	R	RW	RW	R	RW		RW		RW				RW			
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

CMP0_MODE = 0x4001_F010

CMP1_MODE = 0x4002_0010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 16.5. CMPn_MODE Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	INVEN	**Invert Comparator Output Enable.** 0: Do not invert the comparator output. 1: Invert the comparator output.
29:28	Reserved	Must write reset value.
27:26	CMPHYP	**Positive Hysteresis Control.** 00: Disable positive hysteresis. 01: Set positive hysteresis to 5 mV. 10: Set positive hysteresis to 10 mV. 11: Set positive hysteresis to 20 mV.
25:24	CMPHYN	**Negative Hysteresis Control.** 00: Disable negative hysteresis. 01: Set negative hysteresis to 5 mV. 10: Set negative hysteresis to 10 mV. 11: Set negative hysteresis to 20 mV.
23	PWPUEN	**Positive Input Weak Pullup Enable.** 0: Disable the positive input weak pull up. 1: Enable the positive input weak pull up.

Table 16.5. CMPn_MODE Register Bit Descriptions

Bit	Name	Function
22	NWPUEN	**Negative Input Weak Pullup Enable.** 0: Disable the negative input weak pull up. 1: Enable the negative input weak pull up.
21:16	DACLVL	**Comparator DAC Output Level.** These bits control the comparator reference DAC's output voltage. The voltage is given by: $\text{DAC Output} = \text{VREF} \times \left(\frac{\text{DACLVL}}{64}\right)$ where VREF is the positive or negative comparator mux selection, as defined by INMUX.
15	Reserved	Must write reset value.
14	RIEN	**Rising Edge Interrupt Enable.** 0: Disable the comparator rising edge interrupt. 1: Enable the comparator rising edge interrupt.
13	FIEN	**Falling Edge Interrupt Enable.** 0: Disable the comparator falling edge interrupt. 1: Enable the comparator falling edge interrupt.
12	Reserved	Must write reset value.
11:10	CMPMD	**Comparator Mode.** 00: Mode 0 (fastest response time, highest power consumption). 01: Mode 1. 10: Mode 2. 11: Mode 3 (slowest response time, lowest power consumption).
9:8	INMUX	**Input MUX Select.** 00: Connects the NMUX signal to CMP- and the PMUX signal to CMP+. 01: Connects VSS to CMP- and the PMUX signal to CMP+. 10: Connects the NMUX signal to CMP-, the PMUX signal to the Comparator DAC voltage reference, and the DAC output to CMP+. 11: Connects the PMUX signal to CMP+, the NMUX signal to the Comparator DAC voltage reference, and the DAC output to CMP-.

Table 16.5. CMPn_MODE Register Bit Descriptions

Bit	Name	Function
7:4	PMUX	**Positive Input Select.** 0000: Select CMPnP.0. 0001: Select CMPnP.1. 0010: Select CMPnP.2. 0011: Select CMPnP.3. 0100: Select CMPnP.4. 0101: Select CMPnP.5. 0110: Select CMPnP.6. 0111: Select CMPnP.7. 1000: Select CMPnP.8. 1001: Select CMPnP.9. 1010: Select CMPnP.10. 1011: Select CMPnP.11. 1100: Select CMPnP.12. 1101: Select CMPnP.13. 1110: Select CMPnP.14. 1111: Select CMPnP.15.
3:0	NMUX	**Negative Input Select.** 0000: Select CMPnN.0. 0001: Select CMPnN.1. 0010: Select CMPnN.2. 0011: Select CMPnN.3. 0100: Select CMPnN.4. 0101: Select CMPnN.5. 0110: Select CMPnN.6. 0111: Select CMPnN.7. 1000: Select CMPnN.8. 1001: Select CMPnN.9. 1010: Select CMPnN.10. 1011: Select CMPnN.11. 1100: Select CMPnN.12. 1101: Select CMPnN.13. 1110: Select CMPnN.14. 1111: Select CMPnN.15.

16.9. CMPn Register Memory Map

Table 16.6. CMPn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
CMPn_CONTROL	0x0	ALL | SET | CLR	CMPEN	CMPOUT	Reserved															CMPRI	CMPFI	Reserved													
CMPn_MODE	0x10	ALL | SET | CLR	Reserved	INVEN	Reserved		CMPHYP		CMPHYN		PWPUEN	NWPUEN	DACLVL						Reserved	RIEN	FIEN	Reserved	CMPMD		INMUX		PMUX				NMUX			

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: CMP0 = 0x4001_F000, CMP1 = 0x4002_0000

17. DMA Controller (DMACTRL0)

This section describes the DMA Controller (DMACTRL) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the DMACTRL block, which is used by all device families covered in this document.

17.1. DMA Controller Features

The DMA Controller module includes the following features:

- Utilizes ARM PrimeCell uDMA architecture.
- Implements 10 channels.
- DMA crossbar supports direct peripheral data requests and maps peripherals to each channel.
- Supports primary, alternate, and scatter-gather channel transfer structures to implement various types of transfers.
- Access allowed to all APB and AHB memory space.

The DMA facilitates autonomous peripheral operation, allowing the core to finish tasks more quickly without spending time polling or waiting for peripherals to interrupt. This helps reduce the overall power consumption of the system, as the device can spend more time in low-power modes.

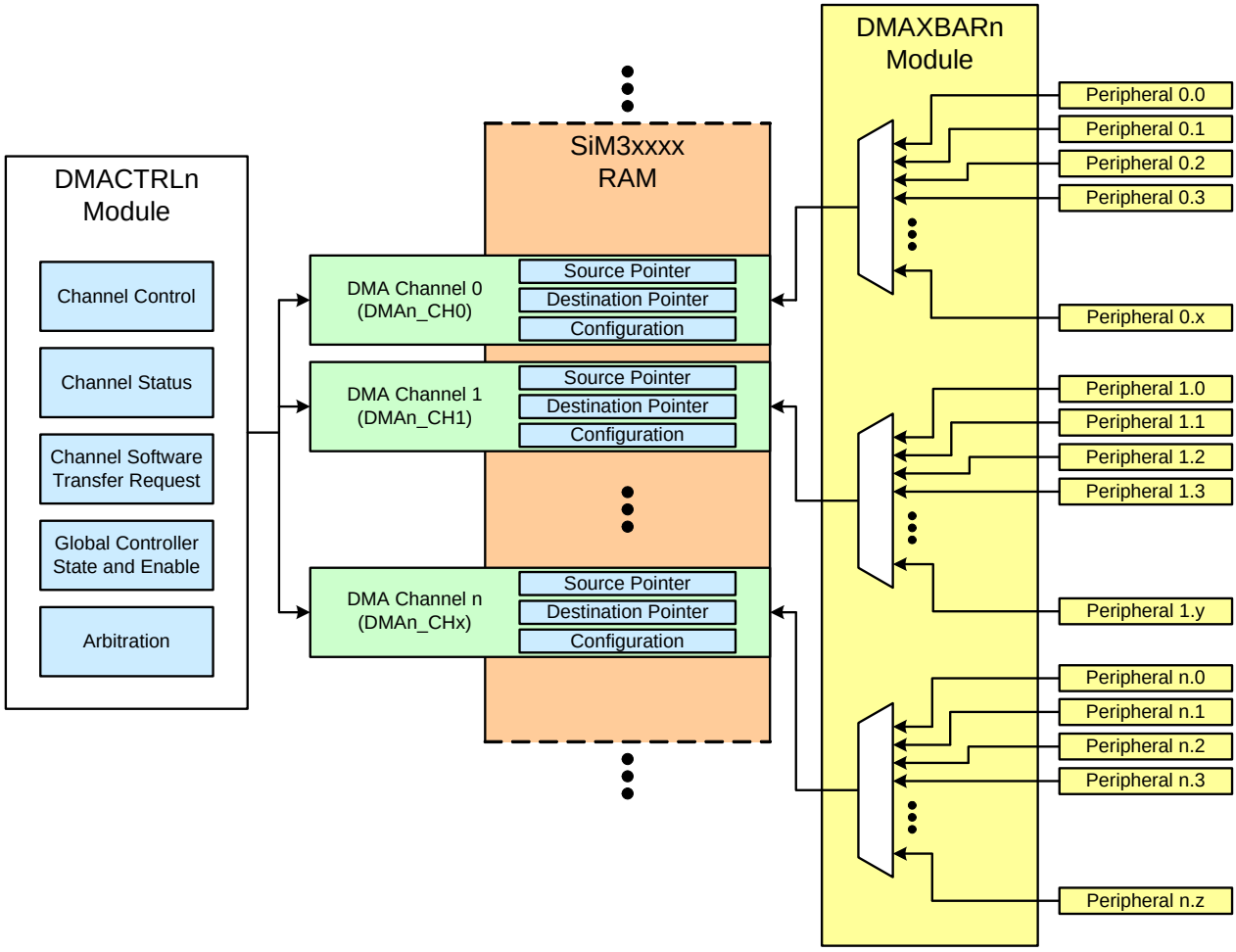


Figure 17.1. DMACTRL and DMACH Block Diagram

17.2. Overview

The DMA controller provides a single access point for all 16 DMA channels and the global DMA controls. The controller is also responsible for handling arbitration between channels.

Each channel has separate enables, alternate enables, masks, software requests, programmable priority, and status flags. The channels operate independently, but have a fixed arbitration order.

The channels have controls and flags in the DMACTRL registers. In addition, each channel has several transfer structures stored in memory that describe the data transfer in detail. Each channel can have a primary, alternate, or scatter-gather structures. The BASEPTR and ABASEPTR registers point to the starting address of these structures in memory. Firmware sets the BASEPTR field, and the controller hardware automatically sets the ABASEPTR field based on the number of channels implemented in the module.

The NUMCHAN field in the STATUS register reports the number of channels implemented on a device. The STATE field reports the current status of the DMA controller, and the DMAENS bit indicates whether the global DMA enable is set.

17.3. Interrupts

Each DMA channel has a separate interrupt vector, and a channel will generate an interrupt at the end of the current transfer. Firmware can enable or disable the interrupts in the NVIC.

17.4. Configuring a DMA Channel

To configure a DMA channel for a data transfer:

1. Enable the DMA module (DMAEN = 1).
2. Set the address location of the channel transfer structures (BASEPTR) according to the restrictions in Table 17.1.
3. Create the primary and any alternate or scatter-gather data structures in memory for the desired transfer.
4. Enable the DMA channel using the CHENSET register.
5. Submit a request to start the transfer.

For software-initiated transfers, a request is issued by setting the channel's software request in the CHSWRCN register. It is recommended that firmware set the channel request mask (CHREQMSET) for channels using software-initiated transfers to avoid any peripherals connected to the channel from requesting DMA transfers.

For peripheral transfers, firmware should configure the peripheral for the DMA transfer and set the device's DMA crossbar (DMAXBAR) to map a DMA channel to the peripheral. The peripheral will request data as needed. The channel request mask (CHREQMCLR) must be cleared for the channel to use peripheral transfers.

The CHALTSET register can set a DMA channel to use the alternate structure instead of the primary structure. Firmware can use the CHALTCLR register to set the channel back to the primary structure. The controller automatically updates the CHALTSET fields to indicate which structure is in use during transfers that use the alternate structure (ping-pong and scatter-gather).

17.5. DMA Channel Transfer Structures

Each channel has transfer structures stored in memory that describe the data transfer in detail. Each structure is composed of four 32-bit words in memory organized as follows:

1. Source End Pointer (word 1): The address of the last source data in the transfer.
2. Destination End Pointer (word 2): The last destination address of the transfer.
3. Channel Configuration (word 3): Configuration details for the transfer.
4. Alignment padding (word 4): Not used by the DMA controller. Firmware my use this word for any purpose.

Each channel can have a primary, alternate, and scatter-gather structure. The primary and alternate structures are organized in contiguous blocks in memory for each of the channels. The spacing for these structures is fixed, so any unused channels must still be accounted for when placing structures in memory. In addition to the fixed structure, the base address (BASEPTR) supports between 22- and 27-bit addresses, depending on the number of channels implemented. The primary structures must be placed at the start of an address block sized for both the primary and alternate structures. Table 17.1 shows the valid base pointer addresses for each range of implemented channels.

The scatter-gather structures are more flexible and can appear anywhere in memory.

Table 17.1. Valid Base Pointer Addresses

Number of Channels Implemented	Base Pointer Size (BASEPTR)	Valid Primary Channel Transfer Structure Addresses	Required Number of Bytes (Primary and Alternate)
1	27 bits [31:5]	multiples of 16 (0x00000010)	32
2	26 bits [31:6]	multiples of 32 (0x00000020)	64
3-4	25 bits [31:7]	multiples of 64 (0x00000040)	128
5-8	24 bits [31:8]	multiples of 128 (0x00000080)	256
9-16	23 bits [31:9]	multiples of 256 (0x00000100)	512
17-32	22 bits [31:10]	multiples of 512 (0x00000200)	1024

Channel 0's primary structure begins at address offset 0x0000, Channel 1's primary structure starts at offset 0x0010, and so on. The alternate structures begin after the last primary structure location for the number of implemented channels, regardless of whether or not the channels are in use.

Firmware originally sets the channel configuration descriptor; the DMA controller will modify this word as the transfer progresses, so firmware should not access this descriptor until any active transfers for the channel complete.

Figure 17.2 shows the fixed memory configuration for the structures.

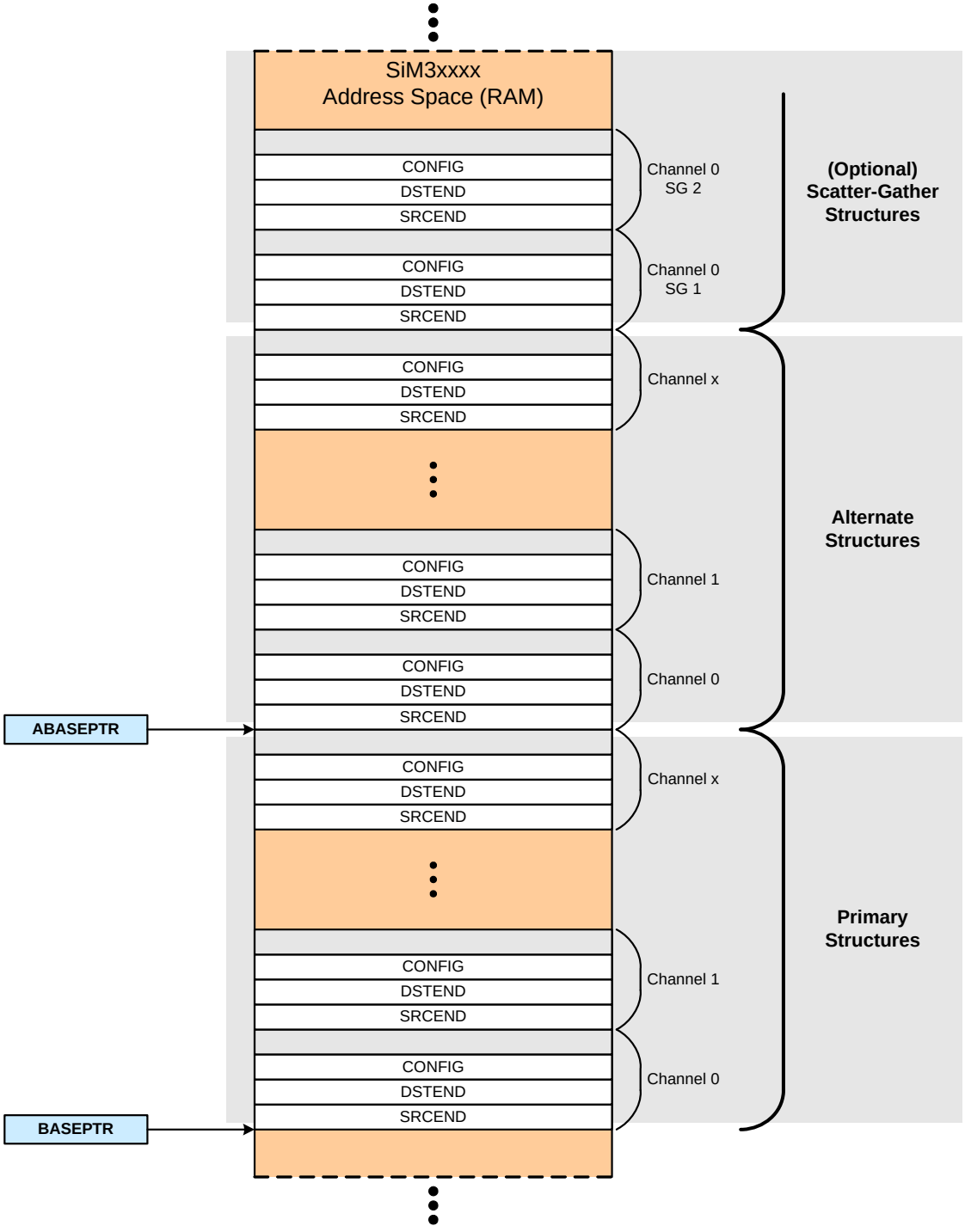


Figure 17.2. Channel Transfer Structure Memory Configuration

17.5.1. Channel Transfer Structure Descriptors

Table 17.2, Table 17.3, Table 17.4 describe the source end pointer, destination pointer, and configuration descriptors for the primary, alternate, and scatter-gather DMA channel structures.

Table 17.2. DMA0_CHx_SRCEND: Source End Pointer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SRCEND[31:16]															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SRCEND[15:0]															

Address in Channel Transfer Structure: 0x0000

Bit	Name	Function
31:0	SRCEND	**Source End Pointer.** This field is the address of the last source data in the DMA transfer.

Table 17.3. DMA0_CHx_DSTEND: Destination End Pointer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DSTEND[31:16]															
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DSTEND[15:0]															

Address in Channel Transfer Structure: 0x0004

Bit	Name	Function
31:0	DSTEND	**Destination End Pointer.** This field is the last destination address of the DMA transfer.

Table 17.4. DMA0_CHx_CONFIG: Channel Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DSTAIMD		DSTSIZE		SRCAIMD		SRCSIZE		Reserved						RPOWER[3:2]	
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RPOWER[1:0]		NCOUNT										Reserved	TMD		

Address in Channel Transfer Structure: 0x0008

Bit	Name	Function
31:30	DSTAIMD	**Destination Address Increment Mode.** This field must be set to a value that's equal to or greater than the DSTSIZE setting. 00: The destination address increments by one byte after each data transfer. 01: The destination address increments by one half-word after each data transfer. 10: The destination address increments by one word after each data transfer. 11: The destination address does not increment.
29:28	DSTSIZE	**Destination Data Size Select.** The destination size (DSTSIZE) must equal the source size (SRCSIZE). 00: Each DMA destination data transfer writes a byte. 01: Each DMA destination data transfer writes a half-word. 10: Each DMA destination data transfer writes a word. 11: Reserved.
27:26	SRCAIMD	**Source Address Increment Mode.** This field must be set to a value that's equal to or greater than the SRCSIZE setting. 00: The source address increments by one byte after each data transfer. 01: The source address increments by one half-word after each data transfer. 10: The source address increments by one word after each data transfer. 11: The source address does not increment.
25:24	SRCSIZE	**Source Data Size Select.** The destination size (DSTSIZE) must equal the source size (SRCSIZE). 00: Each DMA source data transfer reads a byte. 01: Each DMA source data transfer reads a half-word. 10: Each DMA source data transfer reads a word. 11: Reserved.
23:18	Reserved	Must write 0 to this field.

Bit	Name	Function
17:14	RPOWER	**Transfer Size Select.** This field determines the number of data transfers between each DMA channel re-arbitration. The number of data transfers is given by: $\text{Number of Transfers} = 2^{\text{RPOWER}}$ This field is ignored for peripherals that support single data requests only. A value of 0 for RPOWER should be used for channels interfacing with these types of peripherals.
13:4	NCOUNT	**Transfer Total.** This field is the total number of transfers for the DMA channel. The total number is NCOUNT + 1, so software requiring a total of 4 transfers would set the NCOUNT field to 3. The DMA controller decrements this field as transfers are made.
3	Reserved	Must write 0 to this bit.
2:0	TMD	**Transfer Mode.** 000: Stop the DMA channel. 001: Use the Basic transfer type (single structure only). 010: Use the Auto-Request transfer type (single structure only). 011: Use the Ping-Pong transfer type (primary and alternate structures). 100: Use the Memory Scatter-Gather Primary transfer type (primary, alternate, and scattered structures). 101: Use the Memory Scatter-Gather Alternate transfer type (primary, alternate, and scattered structures). 110: Use the Peripheral Scatter-Gather Primary transfer type (primary, alternate, and scattered structures). 111: Use the Peripheral Scatter-Gather Alternate transfer type (primary, alternate, and scattered structures).

17.6. Transfer Types

The DMA channels support five transfer types: basic, auto-request, ping-pong, memory scatter-gather, and peripheral scatter-gather. Table 17.5 shows the memory requirements for each transfer type.

Table 17.5. Transfer Memory Requirements

Transfer Type	Transfer Structures Required			Memory (RAM) Required (bytes)		
	Primary	Alternate	Scatter-Gather	5-8 Channels Implemented	9-16 Channels Implemented	17-32 Channels Implemented
Basic	✓			128	256	512
Auto-Request	✓			128	256	512
Ping-Pong	✓	✓		256	512	1024
Memory Scatter-Gather	✓	✓	✓	256 + SG	512 + SG	1024 + SG
Peripheral Scatter-Gather	✓	✓	✓	256 + SG	512 + SG	1024 + SG

17.6.1. Basic Transfers

The basic transfer type uses only one structure (primary or alternate). In this mode, the channel will make NCOUNT + 1 data moves in 2^{RPOWER} bursts. Each data request moves one 2^{RPOWER} set of data. The number of requests required for a transfer is:

$$\text{Number of Requests} = \frac{NCOUNT + 1}{2^{RPOWER}}$$

Equation 17.1. Number of Requests for Basic Transfers

Any data remaining can be transferred by firmware or use an extra DMA data request.

After the final data transfer:

1. The DMA channel will write the primary structure TMD field with 0.
2. The primary structure NCOUNT field will contain 0.
3. The controller automatically disables the channel (the channel bit in CHENSET will read 0).

Figure 17.3 illustrates the DMA memory structures for a basic transfer.

This transfer type is recommended for peripheral to memory or memory to peripheral transfers.

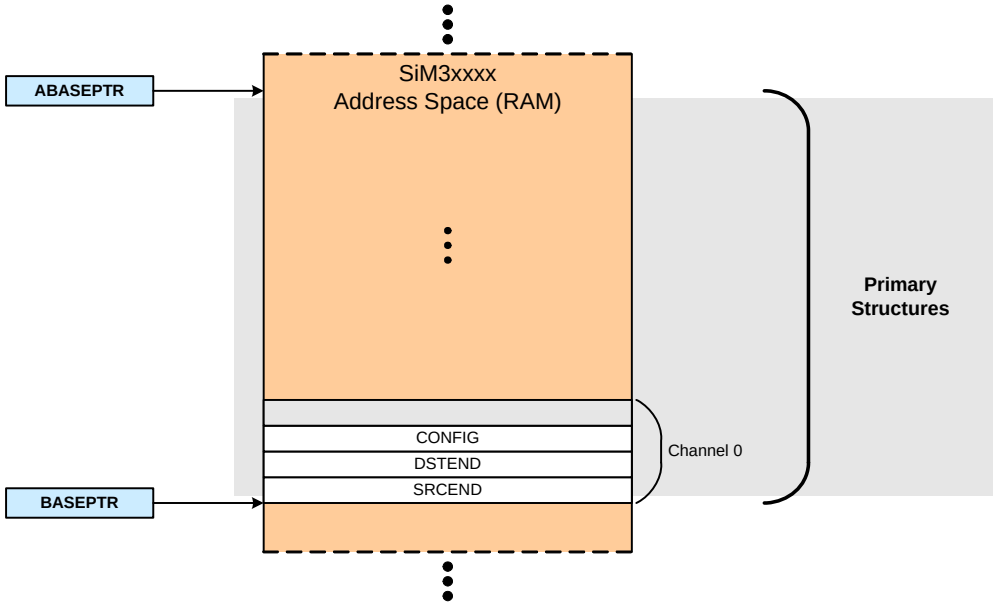


Figure 17.3. Basic and Auto-Request Transfer Memory Configuration

17.6.2. Auto-Request Transfers

Auto-request transfers use only one structure (primary or alternate). This transfer type only requires one data request to transfer all of the data. The controller will arbitrate as normal (every 2^{RPOWER} transfers), and a channel interrupt will occur when the transfer completes. This transfer type is recommended for memory to memory transfers.

After the final data transfer:

1. The DMA channel will write the primary structure TMD field with 0.
2. The primary structure NCOUNT field will contain 0.
3. The controller automatically disables the channel (the channel bit in CHENSET will read 0).

The auto-request memory configuration is identical to the basic transfer shown in Figure 17.3.

17.6.3. Ping-Pong Transfers

Ping-pong transfers use both the primary and alternate channel structures. When the channel completes the transfer described by the first structure, it clears the TMD field in the original structure to 0 and toggles to point to the other structure. A channel interrupt will occur to allow firmware to update the completed transfer's structure, as the ping-pong operation will stop without intervention.

As with basic transfers, each 2^{RPOWER} data moves require a new data request. The number of requests is given by Equation 17.1.

Figure 17.4 shows an example where a channel's primary structure has an RPOWER of 1 with an NCOUNT of 3 and the alternate structure has an RPOWER of 0 with an NCOUNT of 4. These structures are both configured to move words (DSTSIZE and SRCSIZE set to 2) in ping-pong mode (TMD = 3).

Figure 17.5 illustrates the ping-pong memory configuration.

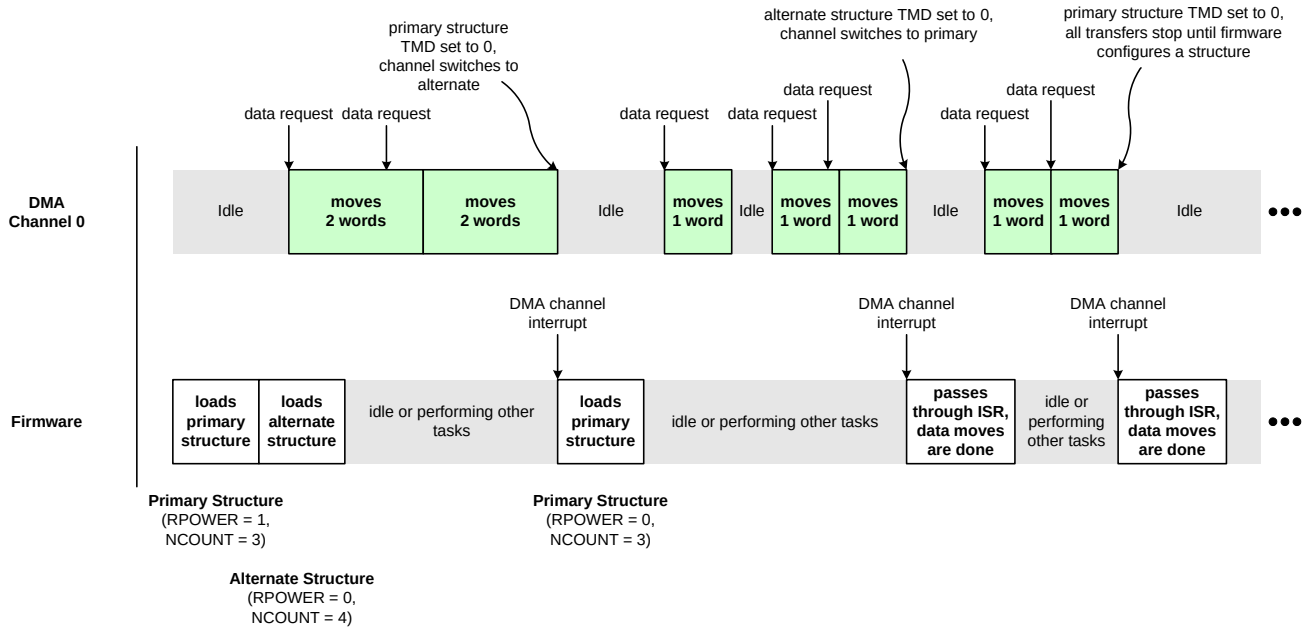


Figure 17.4. Ping-Pong Transfer Example

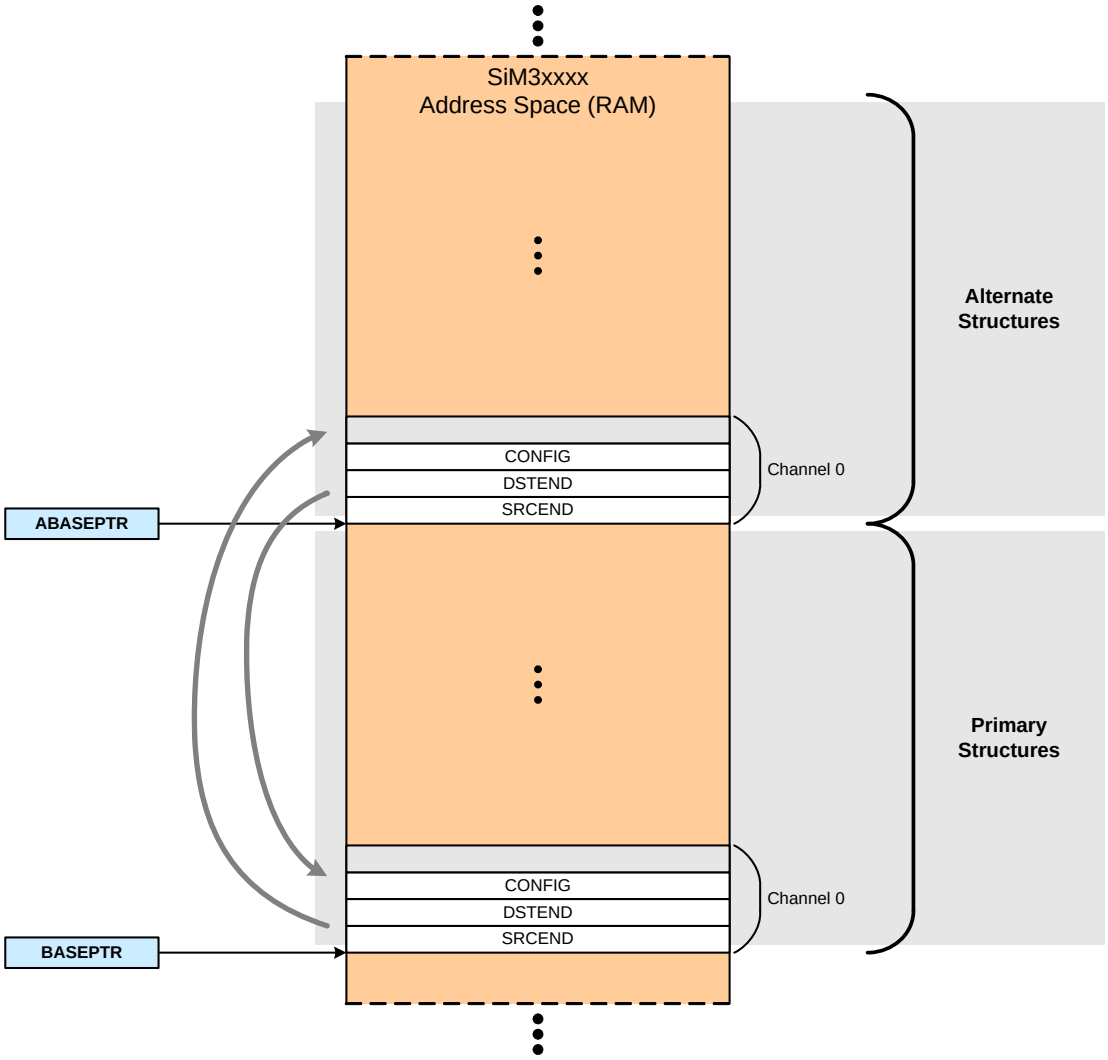


Figure 17.5. Ping-Pong Transfer Memory Configuration

17.6.4. Memory Scatter-Gather Transfers

The memory scatter-gather transfer uses primary, alternate, and scatter-gather structures. This transfer type allows a DMA channel to be set for multiple transfers at once without core intervention at the price of extra memory for the scatter-gather structures.

The primary structure in this mode contains the number and location of the scatter-gather structures. The primary structure should be programmed as follows:

1. Memory scatter-gather primary mode (TMD = 4).
2. RPOWER = 2.
3. NCOUNT set to the value specified by Equation 17.2.
4. SRCEND is set to the location of the last word of all the scatter-gather structures.
5. DSTEND is set to the location of the last word in the single alternate structure.

$$\text{NCOUNT} = (\text{Number of SG Structures} \times 4) - 1$$

Equation 17.2. NCOUNT Value for Scatter-Gather Transfers

The scatter-gather structures must be stacked contiguously in memory. The channel will copy the scatter-gather structures into the alternate structure location and execute them one by one. The scatter-gather structures should be programmed to memory scatter-gather alternate mode (TMD = 5), except for the last structure, which should use the basic or auto-request transfer types (TMD = 1 or 2).

Once started, the DMA channel execution process is as follows:

1. Copy scatter-gather 1 (SG1) to the alternate structure.
2. Jump to the alternate structure and execute.
3. Jump back to the primary structure.
4. Copy scatter-gather 2 (SG2) to the alternate structure.
5. Jump to the alternate structure and execute.
6. Jump back to the primary structure.

The channel will continue in this pattern until the channel encounters a scatter-gather structure set to a basic or auto-request transfer.

Only one data request is required to execute all of the scattered transactions. The channel interrupt will occur once the last scatter-gather structure (programmed to a basic transfer) executes, if enabled. Arbitration occurs every 2^{RPOWER} of the scatter-gather structures.

Figure 17.6 shows the memory scatter-gather memory configuration.

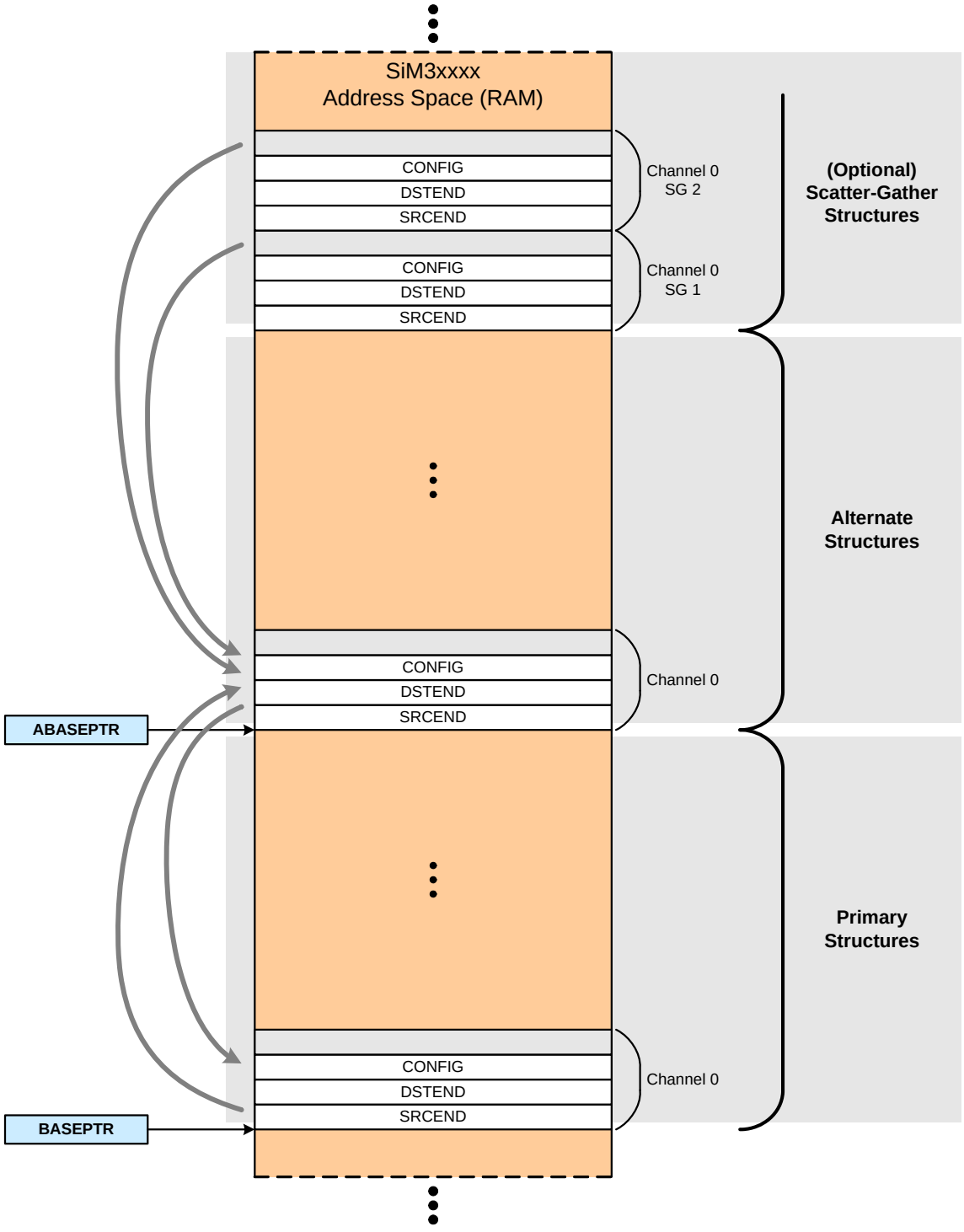


Figure 17.6. Memory and Peripheral Scatter-Gather Transfer Memory Configuration

17.6.5. Peripheral Scatter-Gather Transfers

The peripheral scatter-gather transfer is very similar to the memory scatter-gather transfer and uses primary, alternate, and scatter-gather structures. This transfer type allows a DMA channel to be set for multiple transfers at once without core intervention at the price of extra memory for the scatter-gather structures. A data request is required for each 2^{RPOWER} data move of the scatter-gather structure tasks. The RPOWER value can be different for each scatter-gather task. Equation 17.1 describes the total number of data requests required to complete a transfer.

The primary structure in this mode contains the number and location of the scatter-gather structures. The primary structure should be programmed as follows:

1. Peripheral scatter-gather primary mode (TMD = 6).
2. RPOWER = 2.
3. NCOUNT set to the value specified by Equation 17.2.
4. SRCEND is set to the location of the last word of all the scatter-gather structures.
5. DSTEND is set to the location of the last word in the single alternate structure.

The scatter-gather structures must be stacked contiguously in memory. The channel will copy the scatter-gather structures into the alternate structure location and execute them one by one. The scatter-gather structures should be programmed to peripheral scatter-gather alternate mode (TMD = 7), except for the last structure, which should use the basic or auto-request transfer types (TMD = 1 or 2).

Once started, the DMA channel execution process is as follows:

1. Copy scatter-gather 1 (SG1) to the alternate structure.
2. Jump to the alternate structure and execute.
3. Jump back to the primary structure.
4. Copy scatter-gather 2 (SG2) to the alternate structure.
5. Jump to the alternate structure and execute.
6. Jump back to the primary structure.

The channel will continue in this pattern until the channel encounters a scatter-gather structure set to a basic or auto-request transfer.

The channel interrupt will occur once the last scatter-gather structure (programmed to a basic transfer) executes, if enabled.

Figure 17.6 shows the peripheral scatter-gather memory configuration.

17.7. Data Requests

Each DMA channel has two data requests: single and burst. Peripherals can support single requests, burst requests, or both. If configured to use a DMA channel, peripherals request data as needed using the appropriate request type. Table 17.6 lists the supported requests for the supported triggers and peripherals.

The RPOWER field is only valid for peripherals that support burst requests. For peripherals that only support single requests, the RPOWER field is ignored and re-arbitration occurs after every single data move.

Table 17.6. Supported Trigger or Peripheral Data Requests

Peripheral Module	Supported Request Types
AESn	burst only
EPCAn	burst only
I2Cn	single only
ENCDECn	single only
DTMn	burst only
IDACn	single only
SARADCn	burst only
SPIn	burst only
TIMERn overflow	burst only
USARTn	single only
External Trigger	burst only
Software Trigger	burst only

In addition to peripheral-initiated transfers, all of the supported DMA channels can select the rising or falling edges of one of the DMA external transfer start signals to initiate data transfers. When the selected edge occurs on the external signal, the DMA channels with the DMA0T0/1 signals selected in the DMAXBARx.CHANnSEL field will start the corresponding channel's data transfer as defined by the DMA channel data structure in memory. The DMA module external trigger sources are routed to peripheral pins using the crossbar.

17.8. Masking Channels

DMA channels can be temporarily disabled by setting the channel bit in CHREQMSET. Setting this bit to 1 causes the DMA channel to no longer respond to data requests from peripherals. The channel will always respond to software-initiated transfer requests, even if CHREQMSET is set for the channel. Firmware can write a 1 to the CHREQMCLR register to clear the mask for a channel.

It is recommended that firmware set the channel request mask (CHREQMSET) for channels using software-initiated transfers to avoid any peripherals connected to the channel from requesting DMA transfers.

17.9. Errors

The ERROR bit in the BERRCLR register indicates when a DMA bus error occurs. If enabled, this bit will generate an interrupt.

17.10. Arbitration

The DMA controller is a master on the AHB bus. This allows the module to control data transfers without any interaction with the core.

The channels are in a fixed priority order. Channel 0 has the highest priority, and the last implemented channel has the lowest priority. This fixed order can be superceded by using the programmable high priority setting (CHHPSET). At each re-arbitration period, the controller gives control of the bus to the highest priority channel with a pending data request.

The RPOWER field in the channel transfer structures determines when the re-arbitration periods occur. The channel in control of the bus will make 2^{RPOWER} data moves before the controller re-arbitrates. If the channel still has the highest priority, it can transfer again until the next re-arbitration period. The RPOWER field is only valid for peripherals that support burst requests. For peripherals that only support single requests, re-arbitration will occur after each single data move.

Figure 17.7 shows an example controller arbitration with two channels active. Channel 0 has an RPOWER of 1 (2 data moves), and channel 1 has an RPOWER of 0 (1 data move). Both channels are set to move words (DSTSIZE and SRCSIZE set to 2).

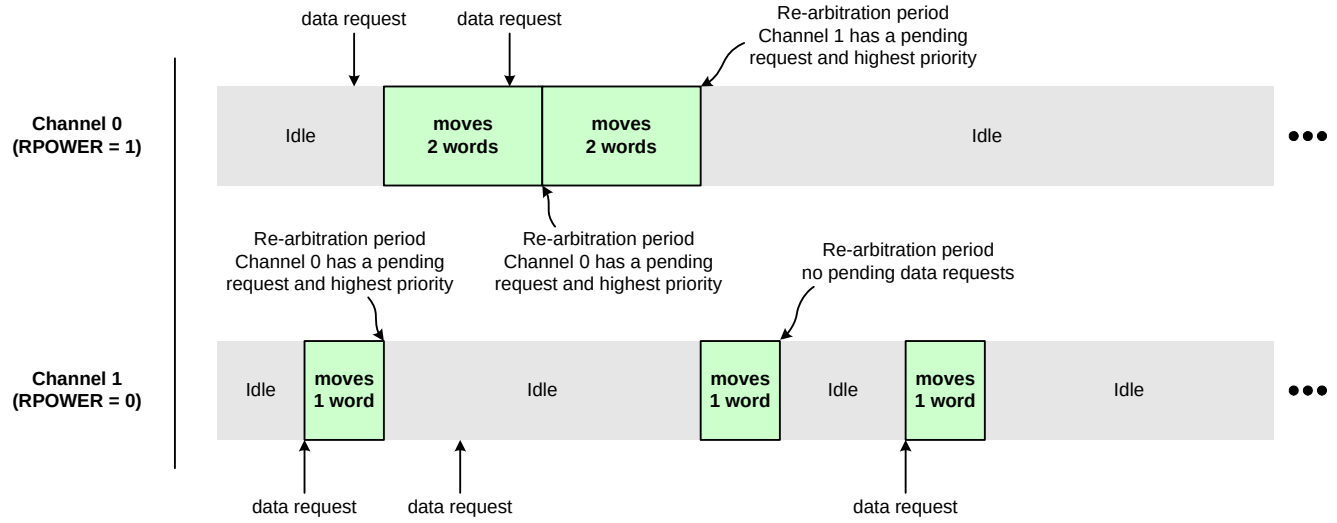


Figure 17.7. DMA Arbitration Example

17.11. DMACTRL0 Registers

This section contains the detailed register descriptions for DMACTRL0 registers.

Register 17.1. DMACTRL0_STATUS: Controller Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved											NUMCHAN				
Type	R											R				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								STATE				Reserved			DMAENSTS
Type	R								R				R			R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_STATUS = 0x4003_6000

Table 17.7. DMACTRL0_STATUS Register Bit Descriptions

Bit	Name	Function
31:21	Reserved	Must write reset value.
20:16	NUMCHAN	**Number of Supported DMA Channels.** This value represents one less than the number of supported DMA channels on the device. For example, a value of 15 in this field means 16 channels are supported on the device.
15:8	Reserved	Must write reset value.
7:4	STATE	**State Machine State.** This field indicates the current state of the control state machine. 0000: Idle. 0001: Reading channel controller data. 0010: Reading source data end pointer. 0011: Reading destination data end pointer. 0100: Reading source data. 0101: Writing destination data. 0110: Waiting for a DMA request to clear. 0111: Writing channel controller data. 1000: Stalled. 1001: Done. 1010: Peripheral scatter-gather transition. 1011-1111: Reserved.

Table 17.7. DMACTRL0_STATUS Register Bit Descriptions

Bit	Name	Function
3:1	Reserved	Must write reset value.
0	DMAENSTS	**DMA Enable Status.** 0: DMA controller is disabled 1: DMA controller is enabled.

Register 17.2. DMACTRL0_CONFIG: Controller Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															DMAEN
Type	W															W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CONFIG = 0x4003_6004

Table 17.8. DMACTRL0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:1	Reserved	Must write reset value.
0	DMAEN	**DMA Enable.** 0: Disable the DMA controller. 1: Enable the DMA controller.

Register 17.3. DMACTRL0_BASEPTR: Base Pointer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BASEPTR[22:7]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BASEPTR[6:0]							Reserved								
Type	RW							R								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_BASEPTR = 0x4003_6008

Table 17.9. DMACTRL0_BASEPTR Register Bit Descriptions

Bit	Name	Function
31:9	BASEPTR	**Control Base Pointer.** This field points to the base location in memory for the DMA channel control data structure. Bits [8:0] are read-only and will always be set to 0. The primary structures must be aligned on a 512 byte boundary..
8:0	Reserved	Must write reset value.

Register 17.4. DMACTRL0_ABASEPTR: Alternate Base Pointer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ABASEPTR[31:16]															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ABASEPTR[15:0]															
Type	R															
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_ABASEPTR = 0x4003_600C

Table 17.10. DMACTRL0_ABASEPTR Register Bit Descriptions

Bit	Name	Function
31:0	ABASEPTR	**Alternate Control Base Pointer.** This read-only field points to the base location in memory for the alternate DMA channel control data structure.

Register 17.5. DMACTRL0_CHSTATUS: Channel Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	R						R	R	R	R	R	R	R	R	R	R
Reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

DMACTRL0_CHSTATUS = 0x4003_6010

Table 17.11. DMACTRL0_CHSTATUS Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Status.** 0: DMA Channel 9 is not waiting for a data request. 1: DMA Channel 9 is waiting for a data request.
8	CH8	**Channel 8 Status.** 0: DMA Channel 8 is not waiting for a data request. 1: DMA Channel 8 is waiting for a data request.
7	CH7	**Channel 7 Status.** 0: DMA Channel 7 is not waiting for a data request. 1: DMA Channel 7 is waiting for a data request.
6	CH6	**Channel 6 Status.** 0: DMA Channel 6 is not waiting for a data request. 1: DMA Channel 6 is waiting for a data request.
5	CH5	**Channel 5 Status.** 0: DMA Channel 5 is not waiting for a data request. 1: DMA Channel 5 is waiting for a data request.
4	CH4	**Channel 4 Status.** 0: DMA Channel 4 is not waiting for a data request. 1: DMA Channel 4 is waiting for a data request.
3	CH3	**Channel 3 Status.** 0: DMA Channel 3 is not waiting for a data request. 1: DMA Channel 3 is waiting for a data request.

Table 17.11. DMACTRL0_CHSTATUS Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Status.** 0: DMA Channel 2 is not waiting for a data request. 1: DMA Channel 2 is waiting for a data request.
1	CH1	**Channel 1 Status.** 0: DMA Channel 1 is not waiting for a data request. 1: DMA Channel 1 is waiting for a data request.
0	CH0	**Channel 0 Status.** 0: DMA Channel 0 is not waiting for a data request. 1: DMA Channel 0 is waiting for a data request.

Register 17.6. DMACTRL0_CHSWRCN: Channel Software Request Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	W						W	W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHSWRCN = 0x4003_6014

Table 17.12. DMACTRL0_CHSWRCN Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Software Request.** 0: DMA Channel 9 does not generate a software data request. 1: DMA Channel 9 generates a software data request.
8	CH8	**Channel 8 Software Request.** 0: DMA Channel 8 does not generate a software data request. 1: DMA Channel 8 generates a software data request.
7	CH7	**Channel 7 Software Request.** 0: DMA Channel 7 does not generate a software data request. 1: DMA Channel 7 generates a software data request.
6	CH6	**Channel 6 Software Request.** 0: DMA Channel 6 does not generate a software data request. 1: DMA Channel 6 generates a software data request.
5	CH5	**Channel 5 Software Request.** 0: DMA Channel 5 does not generate a software data request. 1: DMA Channel 5 generates a software data request.
4	CH4	**Channel 4 Software Request.** 0: DMA Channel 4 does not generate a software data request. 1: DMA Channel 4 generates a software data request.
3	CH3	**Channel 3 Software Request.** 0: DMA Channel 3 does not generate a software data request. 1: DMA Channel 3 generates a software data request.

Table 17.12. DMACTRL0_CHSWRCN Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Software Request.** 0: DMA Channel 2 does not generate a software data request. 1: DMA Channel 2 generates a software data request.
1	CH1	**Channel 1 Software Request.** 0: DMA Channel 1 does not generate a software data request. 1: DMA Channel 1 generates a software data request.
0	CH0	**Channel 0 Software Request.** 0: DMA Channel 0 does not generate a software data request. 1: DMA Channel 0 generates a software data request.

Register 17.7. DMACTRL0_CHREQMSET: Channel Request Mask Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	RW						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHREQMSET = 0x4003_6020

Table 17.13. DMACTRL0_CHREQMSET Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Request Mask Enable.** Read: 0: DMA Channel 9 peripheral data requests enabled. 1: DMA Channel 9 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 9 peripheral data requests.
8	CH8	**Channel 8 Request Mask Enable.** Read: 0: DMA Channel 8 peripheral data requests enabled. 1: DMA Channel 8 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 8 peripheral data requests.
7	CH7	**Channel 7 Request Mask Enable.** Read: 0: DMA Channel 7 peripheral data requests enabled. 1: DMA Channel 7 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 7 peripheral data requests.

Table 17.13. DMACTRL0_CHREQMSET Register Bit Descriptions

Bit	Name	Function
6	CH6	**Channel 6 Request Mask Enable.** Read: 0: DMA Channel 6 peripheral data requests enabled. 1: DMA Channel 6 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 6 peripheral data requests.
5	CH5	**Channel 5 Request Mask Enable.** Read: 0: DMA Channel 5 peripheral data requests enabled. 1: DMA Channel 5 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 5 peripheral data requests.
4	CH4	**Channel 4 Request Mask Enable.** Read: 0: DMA Channel 4 peripheral data requests enabled. 1: DMA Channel 4 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 4 peripheral data requests.
3	CH3	**Channel 3 Request Mask Enable.** Read: 0: DMA Channel 3 peripheral data requests enabled. 1: DMA Channel 3 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 3 peripheral data requests.
2	CH2	**Channel 2 Request Mask Enable.** Read: 0: DMA Channel 2 peripheral data requests enabled. 1: DMA Channel 2 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 2 peripheral data requests.
1	CH1	**Channel 1 Request Mask Enable.** Read: 0: DMA Channel 1 peripheral data requests enabled. 1: DMA Channel 1 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 1 peripheral data requests.

Table 17.13. DMACTRL0_CHREQMSET Register Bit Descriptions

Bit	Name	Function
0	CH0	**Channel 0 Request Mask Enable.** Read: 0: DMA Channel 0 peripheral data requests enabled. 1: DMA Channel 0 peripheral data requests disabled. Write: 0: No effect (use CHREQMCLR to clear). 1: Disable DMA Channel 0 peripheral data requests.

Register 17.8. DMACTRL0_CHREQMCLR: Channel Request Mask Clear

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	W						W	W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHREQMCLR = 0x4003_6024

Table 17.14. DMACTRL0_CHREQMCLR Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 9 peripheral data requests.
8	CH8	**Channel 8 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 8 peripheral data requests.
7	CH7	**Channel 7 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 7 peripheral data requests.
6	CH6	**Channel 6 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 6 peripheral data requests.
5	CH5	**Channel 5 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 5 peripheral data requests.
4	CH4	**Channel 4 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 4 peripheral data requests.
3	CH3	**Channel 3 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 3 peripheral data requests.

Table 17.14. DMACTRL0_CHREQMCLR Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 2 peripheral data requests.
1	CH1	**Channel 1 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 1 peripheral data requests.
0	CH0	**Channel 0 Request Mask Disable.** 0: No effect. 1: Enable DMA Channel 0 peripheral data requests.

Register 17.9. DMACTRL0_CHENSET: Channel Enable Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	R						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHENSET = 0x4003_6028

Table 17.15. DMACTRL0_CHENSET Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Enable.** Read: 0: DMA Channel 9 disabled. 1: DMA Channel 9 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 9.
8	CH8	**Channel 8 Enable.** Read: 0: DMA Channel 8 disabled. 1: DMA Channel 8 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 8.

Notes:

1. The controller will automatically disable a channel when: 1) the controller completes the DMA cycle, 2) it reads a channel configuration memory location and the cycle control field is 0, or 3) an error (ERROR = 1) occurs on the AHB bus.

Table 17.15. DMACTRL0_CHENSET Register Bit Descriptions

Bit	Name	Function
7	CH7	**Channel 7 Enable.** Read: 0: DMA Channel 7 disabled. 1: DMA Channel 7 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 7.
6	CH6	**Channel 6 Enable.** Read: 0: DMA Channel 6 disabled. 1: DMA Channel 6 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 6.
5	CH5	**Channel 5 Enable.** Read: 0: DMA Channel 5 disabled. 1: DMA Channel 5 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 5.
4	CH4	**Channel 4 Enable.** Read: 0: DMA Channel 4 disabled. 1: DMA Channel 4 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 4.
3	CH3	**Channel 3 Enable.** Read: 0: DMA Channel 3 disabled. 1: DMA Channel 3 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 3.

Notes:

1. The controller will automatically disable a channel when: 1) the controller completes the DMA cycle, 2) it reads a channel configuration memory location and the cycle control field is 0, or 3) an error (ERROR = 1) occurs on the AHB bus.

Table 17.15. DMACTRL0_CHENSET Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Enable.** Read: 0: DMA Channel 2 disabled. 1: DMA Channel 2 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 2.
1	CH1	**Channel 1 Enable.** Read: 0: DMA Channel 1 disabled. 1: DMA Channel 1 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 1.
0	CH0	**Channel 0 Enable.** Read: 0: DMA Channel 0 disabled. 1: DMA Channel 0 enabled. Write: 0: No effect (use CHENCLR to clear). 1: Enable DMA Channel 0.

Notes:

1. The controller will automatically disable a channel when: 1) the controller completes the DMA cycle, 2) it reads a channel configuration memory location and the cycle control field is 0, or 3) an error (ERROR = 1) occurs on the AHB bus.

Register 17.10. DMACTRL0_CHENCLR: Channel Enable Clear

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	W						W	W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHENCLR = 0x4003_602C

Table 17.16. DMACTRL0_CHENCLR Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Disable.** 0: No effect. 1: Disable DMA Channel 9.
8	CH8	**Channel 8 Disable.** 0: No effect. 1: Disable DMA Channel 8.
7	CH7	**Channel 7 Disable.** 0: No effect. 1: Disable DMA Channel 7.
6	CH6	**Channel 6 Disable.** 0: No effect. 1: Disable DMA Channel 6.
5	CH5	**Channel 5 Disable.** 0: No effect. 1: Disable DMA Channel 5.
4	CH4	**Channel 4 Disable.** 0: No effect. 1: Disable DMA Channel 4.

Notes:

1. The controller will automatically disable a channel by setting the appropriate bit when: 1) the controller completes the DMA cycle, 2) it reads a channel configuration memory location and the cycle control field is 0, or 3) an error (ERROR = 1) occurs on the AHB bus.

Table 17.16. DMACTRL0_CHENCLR Register Bit Descriptions

Bit	Name	Function
3	CH3	**Channel 3 Disable.** 0: No effect. 1: Disable DMA Channel 3.
2	CH2	**Channel 2 Disable.** 0: No effect. 1: Disable DMA Channel 2.
1	CH1	**Channel 1 Disable.** 0: No effect. 1: Disable DMA Channel 1.
0	CH0	**Channel 0 Disable.** 0: No effect. 1: Disable DMA Channel 0.

Notes:

1. The controller will automatically disable a channel by setting the appropriate bit when: 1) the controller completes the DMA cycle, 2) it reads a channel configuration memory location and the cycle control field is 0, or 3) an error (ERROR = 1) occurs on the AHB bus.

Register 17.11. DMACTRL0_CHALTSET: Channel Alternate Select Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	R						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHALTSET = 0x4003_6030

Table 17.17. DMACTRL0_CHALTSET Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Alternate Enable.** Read: 0: DMA Channel 9 is using primary data structure. 1: DMA Channel 9 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 9.
8	CH8	**Channel 8 Alternate Enable.** Read: 0: DMA Channel 8 is using primary data structure. 1: DMA Channel 8 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 8.

Notes:

1. The controller toggles the value of the channel bit after it completes: 1) the four transfers that the primary data structure specifies for a memory scatter-gather or peripheral scatter-gather DMA cycle, 2) all the transfers that the primary data structure specifies for a ping-pong DMA cycle, or 3) all the transfers that the alternate data structure specifies for the following DMA cycle types (ping-pong, memory scatter-gather, peripheral scatter-gather).

Table 17.17. DMACTRL0_CHALTSET Register Bit Descriptions

Bit	Name	Function
7	CH7	**Channel 7 Alternate Enable.** Read: 0: DMA Channel 7 is using primary data structure. 1: DMA Channel 7 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 7.
6	CH6	**Channel 6 Alternate Enable.** Read: 0: DMA Channel 6 is using primary data structure. 1: DMA Channel 6 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 6.
5	CH5	**Channel 5 Alternate Enable.** Read: 0: DMA Channel 5 is using primary data structure. 1: DMA Channel 5 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 5.
4	CH4	**Channel 4 Alternate Enable.** Read: 0: DMA Channel 4 is using primary data structure. 1: DMA Channel 4 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 4.
3	CH3	**Channel 3 Alternate Enable.** Read: 0: DMA Channel 3 is using primary data structure. 1: DMA Channel 3 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 3.

Notes:

1. The controller toggles the value of the channel bit after it completes: 1) the four transfers that the primary data structure specifies for a memory scatter-gather or peripheral scatter-gather DMA cycle, 2) all the transfers that the primary data structure specifies for a ping-pong DMA cycle, or 3) all the transfers that the alternate data structure specifies for the following DMA cycle types (ping-pong, memory scatter-gather, peripheral scatter-gather).

Table 17.17. DMACTRL0_CHALTSET Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Alternate Enable.** Read: 0: DMA Channel 2 is using primary data structure. 1: DMA Channel 2 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 2.
1	CH1	**Channel 1 Alternate Enable.** Read: 0: DMA Channel 1 is using primary data structure. 1: DMA Channel 1 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 1.
0	CH0	**Channel 0 Alternate Enable.** Read: 0: DMA Channel 0 is using primary data structure. 1: DMA Channel 0 is using alternate data structure. Write: 0: No effect (use CHALTCLR to clear). 1: Use the alternate data structure for DMA Channel 0.

Notes:

1. The controller toggles the value of the channel bit after it completes: 1) the four transfers that the primary data structure specifies for a memory scatter-gather or peripheral scatter-gather DMA cycle, 2) all the transfers that the primary data structure specifies for a ping-pong DMA cycle, or 3) all the transfers that the alternate data structure specifies for the following DMA cycle types (ping-pong, memory scatter-gather, peripheral scatter-gather).

Register 17.12. DMACTRL0_CHALTCLR: Channel Alternate Select Clear

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	W						W	W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHALTCLR = 0x4003_6034

Table 17.18. DMACTRL0_CHALTCLR Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 9.
8	CH8	**Channel 8 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 8.
7	CH7	**Channel 7 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 7.
6	CH6	**Channel 6 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 6.
5	CH5	**Channel 5 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 5.
4	CH4	**Channel 4 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 4.
3	CH3	**Channel 3 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 3.

Table 17.18. DMACTRL0_CHALTCLR Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 2.
1	CH1	**Channel 1 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 1.
0	CH0	**Channel 0 Alternate Disable.** 0: No effect. 1: Use the primary data structure for DMA Channel 0.

Register 17.13. DMACTRL0_CHHPSET: Channel High Priority Set

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	R						RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHHPSET = 0x4003_6038

Table 17.19. DMACTRL0_CHHPSET Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 High Priority Enable.** Read: 0: DMA Channel 9 is using the default priority level. 1: DMA Channel 9 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 9.
8	CH8	**Channel 8 High Priority Enable.** Read: 0: DMA Channel 8 is using the default priority level. 1: DMA Channel 8 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 8.
7	CH7	**Channel 7 High Priority Enable.** Read: 0: DMA Channel 7 is using the default priority level. 1: DMA Channel 7 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 7.

Table 17.19. DMACTRL0_CHHPSET Register Bit Descriptions

Bit	Name	Function
6	CH6	**Channel 6 High Priority Enable.** Read: 0: DMA Channel 6 is using the default priority level. 1: DMA Channel 6 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 6.
5	CH5	**Channel 5 High Priority Enable.** Read: 0: DMA Channel 5 is using the default priority level. 1: DMA Channel 5 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 5.
4	CH4	**Channel 4 High Priority Enable.** Read: 0: DMA Channel 4 is using the default priority level. 1: DMA Channel 4 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 4.
3	CH3	**Channel 3 High Priority Enable.** Read: 0: DMA Channel 3 is using the default priority level. 1: DMA Channel 3 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 3.
2	CH2	**Channel 2 High Priority Enable.** Read: 0: DMA Channel 2 is using the default priority level. 1: DMA Channel 2 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 2.
1	CH1	**Channel 1 High Priority Enable.** Read: 0: DMA Channel 1 is using the default priority level. 1: DMA Channel 1 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 1.

Table 17.19. DMACTRL0_CHHPSET Register Bit Descriptions

Bit	Name	Function
0	CH0	**Channel 0 High Priority Enable.** Read: 0: DMA Channel 0 is using the default priority level. 1: DMA Channel 0 is using the high priority level. Write: 0: No effect (use CHHPCLR to clear). 1: Use the high priority level for DMA Channel 0.

Register 17.14. DMACTRL0_CHHPCLR: Channel High Priority Clear

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CH9	CH8	CH7	CH6	CH5	CH4	CH3	CH2	CH1	CH0
Type	W						W	W	W	W	W	W	W	W	W	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_CHHPCLR = 0x4003_603C

Table 17.20. DMACTRL0_CHHPCLR Register Bit Descriptions

Bit	Name	Function
31:10	Reserved	Must write reset value.
9	CH9	**Channel 9 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 9.
8	CH8	**Channel 8 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 8.
7	CH7	**Channel 7 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 7.
6	CH6	**Channel 6 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 6.
5	CH5	**Channel 5 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 5.
4	CH4	**Channel 4 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 4.
3	CH3	**Channel 3 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 3.

Table 17.20. DMACTRL0_CHHPCLR Register Bit Descriptions

Bit	Name	Function
2	CH2	**Channel 2 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 2.
1	CH1	**Channel 1 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 1.
0	CH0	**Channel 0 High Priority Disable.** 0: No effect. 1: Use the high default level for DMA Channel 0.

Register 17.15. DMACTRL0_BERRCLR: Bus Error Clear

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															ERROR
Type	R															RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMACTRL0_BERRCLR = 0x4003_604C

Table 17.21. DMACTRL0_BERRCLR Register Bit Descriptions

Bit	Name	Function
31:1	Reserved	Must write reset value.
0	ERROR	**DMA Bus Error Clear.** Read: 0: DMA error did not occur. 1: DMA error occurred since the last time ERROR was cleared. Write: 0: No effect. 1: Clear the DMA error flag.

17.12. DMACTRL0 Register Memory Map

Table 17.22. DMACTRL0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DMACTRL0_STATUS	0x4003_6000	ALL	Reserved											NUMCHAN					Reserved								STATE				Reserved			DMAENSTS
DMACTRL0_CONFIG	0x4003_6004	ALL	Reserved																															DMAEN
DMACTRL0_BASEPTR	0x4003_6008	ALL	BASEPTR																							Reserved								

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 17.22. DMACTRL0 Memory Map

Register Name	DMACTRL0_ABASEPTR	DMACTRL0_CHSTATUS	DMACTRL0_CHSWRCN
ALL Address	0x4003_600C	0x4003_6010	0x4003_6014
Access Methods	ALL	ALL	ALL
Bit 31	ABASEPTR	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15			
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9		CH9	CH9
Bit 8		CH8	CH8
Bit 7		CH7	CH7
Bit 6		CH6	CH6
Bit 5		CH5	CH5
Bit 4		CH4	CH4
Bit 3		CH3	CH3
Bit 2		CH2	CH2
Bit 1		CH1	CH1
Bit 0		CH0	CH0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 17.22. DMACTRL0 Memory Map

Register Name	DMACTRL0_CHREQMSET	DMACTRL0_CHREQMCLR	DMACTRL0_CHENSET
ALL Address	0x4003_6020	0x4003_6024	0x4003_6028
Access Methods	ALL	ALL	ALL
Bit 31	Reserved	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15			
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9	CH9	CH9	CH9
Bit 8	CH8	CH8	CH8
Bit 7	CH7	CH7	CH7
Bit 6	CH6	CH6	CH6
Bit 5	CH5	CH5	CH5
Bit 4	CH4	CH4	CH4
Bit 3	CH3	CH3	CH3
Bit 2	CH2	CH2	CH2
Bit 1	CH1	CH1	CH1
Bit 0	CH0	CH0	CH0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 17.22. DMACTRL0 Memory Map

Register Name	DMACTRL0_CHENCLR	DMACTRL0_CHALTSET	DMACTRL0_CHALTCLR
ALL Address	0x4003_602C	0x4003_6030	0x4003_6034
Access Methods	ALL	ALL	ALL
Bit 31	Reserved	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15			
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9	CH9	CH9	CH9
Bit 8	CH8	CH8	CH8
Bit 7	CH7	CH7	CH7
Bit 6	CH6	CH6	CH6
Bit 5	CH5	CH5	CH5
Bit 4	CH4	CH4	CH4
Bit 3	CH3	CH3	CH3
Bit 2	CH2	CH2	CH2
Bit 1	CH1	CH1	CH1
Bit 0	CH0	CH0	CH0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 17.22. DMACTRL0 Memory Map

Register Name	DMACTRL0_CHHPSET	DMACTRL0_CHHPCLR	DMACTRL0_BERRCLR
ALL Address	0x4003_6038	0x4003_603C	0x4003_604C
Access Methods	ALL	ALL	ALL
Bit 31	Reserved	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15			
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9	CH9	CH9	
Bit 8	CH8	CH8	
Bit 7	CH7	CH7	
Bit 6	CH6	CH6	
Bit 5	CH5	CH5	
Bit 4	CH4	CH4	
Bit 3	CH3	CH3	
Bit 2	CH2	CH2	
Bit 1	CH1	CH1	
Bit 0	CH0	CH0	ERROR

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

18. DMA Crossbar (DMAXBAR0)

This section describes the DMA Crossbar, and is applicable to all products in the following device families, unless otherwise stated

- SiM3L1xx

18.1. DMA Crossbar Features

The DMA Crossbar includes the following features:

- Maps peripherals to 10 DMA channels to provide flexibility.
- Default arbitration priority assignment (Channel 0 highest, Channel 9 lowest).

The DMA Crossbar can be used to assign a channel to a particular peripheral. These assignments are shown in Table 18.1.

Table 18.1. DMA Crossbar Channel Peripheral Assignments

Peripheral	DMA Channel 0	DMA Channel 1	DMA Channel 2	DMA Channel 3	DMA Channel 4	DMA Channel 5	DMA Channel 6	DMA Channel 7	DMA Channel 8	DMA Channel 9
DTM0 A	✓									
DTM0 B		✓								
DTM0 C			✓							
DTM0 D				✓						
DTM1 A					✓					
DTM1 B						✓				
DTM1 C							✓			
DTM1 D								✓		
DTM2 A			✓				✓			
DTM2 B				✓				✓		
DTM2 C					✓				✓	
DTM2 D						✓				✓
SPI0 TX	✓				✓				✓	
SPI0 RX		✓				✓				✓
ENCDEC0 TX			✓				✓			
ENCDEC0 RX				✓				✓		
AES0 TX	✓				✓					
AES0 RX		✓				✓				

Table 18.1. DMA Crossbar Channel Peripheral Assignments (Continued)

Peripheral	DMA Channel 0	DMA Channel 1	DMA Channel 2	DMA Channel 3	DMA Channel 4	DMA Channel 5	DMA Channel 6	DMA Channel 7	DMA Channel 8	DMA Channel 9
AES0 XOR			✓				✓			
SPI1 TX				✓				✓		
SPI1RX					✓				✓	
USART0 TX		✓		✓			✓		✓	
USART0 RX	✓		✓			✓		✓		
I2C0 RX	✓		✓	✓		✓	✓		✓	✓
I2C0 TX	✓			✓			✓			✓
SARADC0		✓			✓		✓		✓	
IDAC0			✓			✓		✓		✓
EPCA0 Capture	✓	✓			✓				✓	✓
EPCA0 Control		✓			✓	✓				✓
TIMER0L Overflow	✓		✓		✓		✓		✓	
TIMER0H Overflow	✓		✓		✓		✓		✓	
TIMER1L Overflow		✓		✓		✓		✓		✓
TIMER1H Overflow		✓		✓		✓		✓		✓
DMAXT0	✓		✓		✓		✓		✓	
DMAXT1		✓		✓		✓		✓		✓
Software Trigger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

18.2. Channel Priority

In addition to the default priority order where DMA Channel 0 has the highest priority and DMA Channel 9 has the lowest priority, each channel supports a programmable priority. This priority can be programmed in the DMA Controller module (DMACTRLn). Each peripheral may be assigned at most one DMA channel.

18.3. DMAXBAR0 Registers

This section contains the detailed register descriptions for DMAXBAR0 registers.

Register 18.1. DMAXBAR0_DMAXBAR0: Channel 0-7 Trigger Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CH7SEL				CH6SEL				CH5SEL				CH4SEL			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CH3SEL				CH2SEL				CH1SEL				CH0SEL			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMAXBAR0_DMAXBAR0 = 0x4003_7000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 18.2. DMAXBAR0_DMAXBAR0 Register Bit Descriptions

Bit	Name	Function
31:28	CH7SEL	**DMA Channel 7 Peripheral Select.** 0000: Service DTM1 D data requests. 0001: Service DTM2 B data requests. 0010: Service ENCDEC0 RX data requests. 0011: Service SPI1 TX data requests. 0100: Service USART0 RX data requests. 0101: Service IDAC0 data requests. 0110: Service TIMER1L overflow data requests. 0111: Service TIMER1H overflow data requests. 1000: Service DMA0T1 rising edge data requests. 1001: Service DMA0T1 falling edge data requests. 1010-1110: Reserved. 1111: Unassigned.

Table 18.2. DMAXBAR0_DMAXBAR0 Register Bit Descriptions

Bit	Name	Function
27:24	CH6SEL	**DMA Channel 6 Peripheral Select.** 0000: Service DTM1 C data requests. 0001: Service DTM2 A data requests. 0010: Service ENCDEC0 TX data requests. 0011: Service AES0 XOR data requests. 0100: Service USART0 TX data requests. 0101: Service I2C0 RX data requests. 0110: Service I2C0 TX data requests. 0111: Service SARADC0 data requests. 1000: Service TIMER0L overflow data requests. 1001: Service TIMER0H overflow data requests. 1010: Service DMA0T0 rising edge data requests. 1011: Service DMA0T0 falling edge data requests. 1100-1110: Reserved. 1111: Unassigned.
23:20	CH5SEL	**DMA Channel 5 Peripheral Select.** 0000: Service DTM1 B data requests. 0001: Service DTM2 D data requests. 0010: Service SPI0 RX data requests. 0011: Service AES0 RX data requests. 0100: Service USART0 RX data requests. 0101: Service I2C0 RX data requests. 0110: Service IDAC0 data requests. 0111: Service EPCA0 control data requests. 1000: Service TIMER1L overflow data requests. 1001: Service TIMER1H overflow data requests. 1010: Service DMA0T1 rising edge data requests. 1011: Service DMA0T1 falling edge data requests. 1100-1110: Reserved. 1111: Unassigned.
19:16	CH4SEL	**DMA Channel 4 Peripheral Select.** 0000: Service DTM1 A data requests. 0001: Service DTM2 C data requests. 0010: Service SPI1 TX data requests. 0011: Service AES0 TX data requests. 0100: Service SARADC0 data requests. 0101: Service EPCA0 capture data requests. 0110: Service EPCA0 control data requests. 0111: Service TIMER0L overflow data requests. 1000: Service TIMER0H overflow data requests. 1001: Service DMA0T0 rising edge data requests. 1010: Service DMA0T0 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.

Table 18.2. DMAXBAR0_DMAXBAR0 Register Bit Descriptions

Bit	Name	Function
15:12	CH3SEL	**DMA Channel 3 Peripheral Select.** 0000: Service DTM0 D data requests. 0001: Service DTM2 B data requests. 0010: Service ENCDEC0 RX data requests. 0011: Service SPI1 RX data requests. 0100: Service USART0 TX data requests. 0101: Service I2C0 RX data requests. 0110: Service I2C0 TX data requests. 0111: Service TIMER1L overflow data requests. 1000: Service TIMER1H overflow data requests. 1001: Service DMA0T1 rising edge data requests. 1010: Service DMA0T1 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.
11:8	CH2SEL	**DMA Channel 2 Peripheral Select.** 0000: Service DTM0 C data requests. 0001: Service DTM2 A data requests. 0010: Service ENCDEC0 TX data requests. 0011: Service AES0 XOR data requests. 0100: Service SPI1 TX data requests. 0101: Service USART0 RX data requests. 0110: Service I2C0 RX data requests. 0111: Service IDAC0 data requests. 1000: Service TIMER0L overflow data requests. 1001: Service TIMER0H overflow data requests. 1010: Service DMA0T0 rising edge data requests. 1011: Service DMA0T0 falling edge data requests. 1100-1110: Reserved. 1111: Unassigned.
7:4	CH1SEL	**DMA Channel 1 Peripheral Select.** 0000: Service DTM0 B data requests. 0001: Service SPI0 RX data requests. 0010: Service AES0 RX data requests. 0011: Service USART0 TX data requests. 0100: Service SARADC0 data requests. 0101: Service EPCA0 capture data requests. 0110: Service EPCA0 control data requests. 0111: Service TIMER1L overflow data requests. 1000: Service TIMER1H overflow data requests. 1001: Service DMA0T1 rising edge data requests. 1010: Service DMA0T1 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.

Table 18.2. DMAXBAR0_DMAXBAR0 Register Bit Descriptions

Bit	Name	Function
3:0	CH0SEL	**DMA Channel 0 Peripheral Select.** 0000: Service DTM0 A data requests. 0001: Service SPI0 TX data requests. 0010: Service AES0 TX data requests. 0011: Service USART0 RX data requests. 0100: Service I2C0 RX data requests. 0101: Service I2C0 TX data requests. 0110: Service EPCA0 capture data requests. 0111: Service TIMER0L overflow data requests. 1000: Service TIMER0H overflow data requests. 1001: Service DMA0T0 rising edge data requests. 1010: Service DMA0T0 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.

Register 18.2. DMAXBAR0_DMAXBAR1: Channel 8-15 Trigger Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								CH9SEL				CH8SEL			
Type	R								RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

DMAXBAR0_DMAXBAR1 = 0x4003_7010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 18.3. DMAXBAR0_DMAXBAR1 Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:4	CH9SEL	**DMA Channel 9 Peripheral Select.** 0000: Service DTM2 D data requests. 0001: Service SPI0 TX data requests. 0010: Service I2C0 RX data requests. 0011: Service I2C0 TX data requests. 0100: Service IDAC0 data requests. 0101: Service EPCA0 capture data requests. 0110: Service EPCA0 control data requests. 0111: Service TIMER1L overflow data requests. 1000: Service TIMER1H overflow data requests. 1001: Service DMA0T1 rising edge data requests. 1010: Service DMA0T1 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.

Table 18.3. DMAXBAR0_DMAXBAR1 Register Bit Descriptions

Bit	Name	Function
3:0	CH8SEL	**DMA Channel 8 Peripheral Select.** 0000: Service DTM2 C data requests. 0001: Service SPI0 TX data requests. 0010: Service SPI1 RX data requests. 0011: Service USART0 TX data requests. 0100: Service I2C0 RX data requests. 0101: Service SARADC0 data requests. 0110: Service EPCA0 capture data requests. 0111: Service TIMER0L overflow data requests. 1000: Service TIMER0H overflow data requests. 1001: Service DMA0T0 rising edge data requests. 1010: Service DMA0T0 falling edge data requests. 1011-1110: Reserved. 1111: Unassigned.

18.4. DMAXBAR0 Register Memory Map

Table 18.4. DMAXBAR0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DMAXBAR0_DMAXBAR0	0x4003_7000	ALL | SET | CLR	CH7SEL				CH6SEL				CH5SEL				CH4SEL				CH3SEL				CH2SEL				CH1SEL				CH0SEL			
DMAXBAR0_DMAXBAR1	0x4003_7010	ALL | SET | CLR	Reserved																								CH9SEL				CH8SEL			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

19. Data Transfer Manager (DTM0, DTM1 and DTM2)

This section describes the Data Transfer Manager (DTM) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the DTM block, which is used by all device families covered in this document.

19.1. DTM Features

The DTM module includes the following features:

- State descriptions stored in memory with up to 15 states supported per module.
- Supports up to 15 source peripherals and up to 15 destination peripherals per module, in addition to memory or peripherals that do not require a data request.
- Includes error detection and an optional transfer timeout.
- Includes notifications for state transitions.

19.2. Overview

The Data Transfer Manager (DTM) module collects DMA request signals from various peripherals and generates a series of master DMA requests based on a state-driven configuration. This master request drives a set of DMA channels to perform functions such as assembling and transferring communication packets to external radio peripherals. This capability saves power by allowing the MCU to remain in low power modes such as PM2 during complex transfer operations. A combination of simple and peripheral-scatter-gather DMA configurations can be used to perform complex operations while limiting the memory requirements (for example, by implementing direct peripherals-to-peripheral transfers).

Each DTM block supports up to 15 user-configurable states. Each state can be set up to run a certain number of DMA operations from one peripheral (the source) to another (the destination), including memory areas such as Flash and RAM. Each state also has the ability to define two options for what the next state in the sequence will be, dependent on the condition of the counters and other parameters in the DTM block.

Each DTM block is capable of driving up to four DMA channels (A, B, C and D), and each DTM state can be configured to drive a particular request line to the DMA. This allows basic DMA operations to replace a long sequence of peripheral-scatter-gather tasks in most applications, saving memory.

19.3. Counters

The DTM modules contain three different counters: a master counter, a state counter, and a timeout counter. The 16-bit master counter, represented in the MSTCOUNT register, can be initialized by firmware to track the number of DMA requests that have occurred. MSTCOUNT is decremented on each DMA operation unless the active state configuration specifies otherwise.

The 8-bit State counter, represented by the STCOUNT field in the CONTROL register, also decrements each time a DMA request is generated. This is used to track the number of requests since the active state was last entered (from 1 to 256). The STCOUNT field is automatically loaded with the value of STRELOAD in the state description when a state is entered.

A 16-bit timeout counter is represented by the TOCOUNT field in the TIMEOUT register. An internal 8-bit prescaler divides the APB clock frequency and TOCOUNT is decremented every 256 APB clock cycles. Each state can selectively reload TOCOUNT and enable or disable the timeout counter while the state is active. If a TOCOUNT reload is requested, the timeout counter will be reloaded from the TORELOAD field in the TIMEOUT register, and the internal prescaler will reset. If TOCOUNT reaches 0 and the internal prescaler overflows, a timeout error is declared and the DTM transitions to its DONE state. The TOERRI flag in the CONTROL register will be set and an interrupt will be generated if enabled. When it is used, the length of the timeout is equal to 256 x (TRELOAD + 1) APB clock cycles.

19.4. State Machine Control

Each of the 15 available states in a DTM block has configuration information which defines the state operation when it is active. States are set up by firmware in the RAM or Flash region of the device, and when a state becomes active, its information is read into the DTM block's STATE register.

19.4.1. Source, Destination, and DTM Channel

The SRCMOD and DSTMOD fields define the source trigger and the destination trigger for the transfers that will occur in the active state. The available sources and destinations are detailed in Table 19.1 and Table 19.2. If the required DMA transfer is going to or from a memory location, the value 1111b (0xF) should be used in the corresponding field.

Table 19.1. DTM Source Module Options

SRCMOD	Source	SRCMOD	Source
0000	SPI0 Receive	1000	EPCA0 Capture
0001	SPI1 Receive	1001	ENCDEC0 Output
0010	AES0 Output	1010	Reserved
0011	Reserved	1011	Reserved
0100	USART0 Receive	1100	Reserved
0101	Reserved	1101	DMA0T0
0110	I2C0 Receive	1110	DMA0T1
0111	SARADC0 Output	1111	Memory Transfer (No Source)

Table 19.2. DTM Destination Module Options

DSTMOD	Destination	DSTMOD	Destination
0000	SPI0 Transmit	1000	EPCA0 Capture
0001	SPI1 Transmit	1001	ENCDEC0 Input
0010	AES0 Data In	1010	Reserved
0011	AES0 XOR In	1011	Reserved
0100	USART0 Transmit	1100	Reserved
0101	Reserved	1101	DMA0T0
0110	I2C0 Transmit	1110	DMA0T1
0111	IDAC0 Input	1111	Memory Transfer (No Destination)

When a given state is active, the DTM waits until both its source and destination peripherals have asserted their DMA request signals, indicating that both are ready to transmit/receive DMA traffic. At this time, the DTM asserts its master DMA request signal for the channel specified in the state's DTMCHSEL field, causing the DMA engine to perform the next task in that channel's sequence of operation. This DMA task satisfies the source and destination peripheral requests by moving data from the source to the destination. In general, the source and destination peripherals will not be assigned to a DMA channel in the DMA crossbar, and all related DMA traffic will be requested by the DTM.

19.4.2. State Transitions

Each state is associated with a number, 0 through 14. The number 15 is reserved for a DONE state, which terminates DTM operations. The states define two possible paths for the next state, defined in the PRIST (primary state) and SECST (secondary state) fields of the state structure. These two fields may be loaded with any valid state value, including 15 (the DONE state). A simple representation of the DTM state transitions is shown in Figure 19.1.

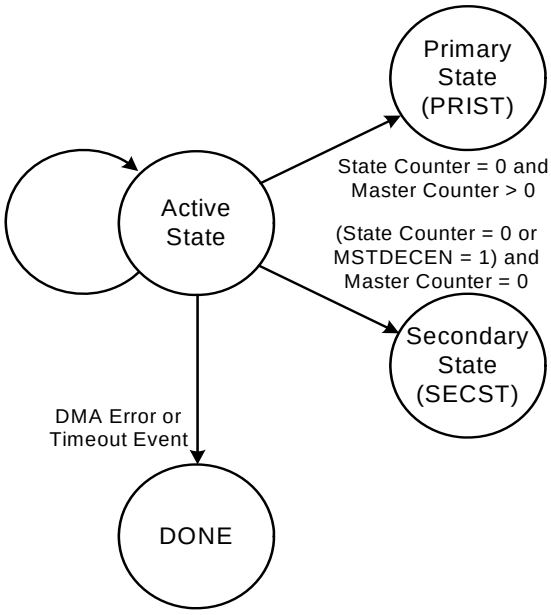


Figure 19.1. State Transition Diagram

When a state is entered, it becomes the active state. Its information is loaded from memory into the STATE register, and its state number will be reported in the ST field of the CONTROL register. At the same time, the state counter (STCOUNT) will be loaded with the value in the state's STRELOAD field. While a state is active, the DTM will manage the data transfer between the selected source and destination peripherals, using the selected DTM channel to request DMA operations. The operation will last as long as the DMA is still actively transferring the data. After the transfer is complete, the state counter is decremented. If the MSTDECEN bit in the state structure is set to 1, the master counter will also be decremented.

If the master counter is non-zero and the state counter is equal to zero, the state machine will transition to the primary state defined by PRIST. If the master counter reaches zero and either the state counter is zero or MSTDECEN = 1, the state machine will transition to the secondary state defined by SECST. Finally, if a timeout error occurs (TOCOUNT reaches zero) when timeouts are enabled, or if a DMA error occurs for the selected channel, the state machine will transition to the DONE state and generate the appropriate flags. Upon exit from a state, the value of that state is loaded into the LASTST field in the CONTROL register.

In some scenarios, a state will need to remain active until MSTCOUNT reaches zero, even if there are more than 256 requests generated. In such cases, this is accomplished by setting the value of PRIST to the active state number.

It is also possible to instruct a state to hold off any further transfer requests until an external pin input (specified by the INHSEL field in the CONTROL register) is asserted. The DTMINH and INHSPOL fields in the state structure configure this capability for the selected inhibit pin. See Table 19.3 for the available pin selections on each package.

Table 19.3. DTM Inhibit Pin Mapping

DTM Inhibit Signal (selected by INHSEL)	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
DTMnINH.0	PB0.0	PB0.0	PB0.0
DTMnINH.1	PB0.1	PB0.1	PB0.1

Table 19.3. DTM Inhibit Pin Mapping

DTM Inhibit Signal (selected by INHSEL)	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
DTMnINH.2	PB0.2	PB0.2	PB0.2
DTMnINH.3	PB0.3	PB0.3	PB0.3
DTMnINH.4	PB0.4	PB0.4	PB0.4
DTMnINH.5	PB0.5	PB0.5	PB0.5
DTMnINH.6	PB0.6	PB0.6	PB0.6
DTMnINH.7	PB0.7	PB0.7	PB0.7
DTMnINH.8	PB2.0	PB2.0	PB2.0
DTMnINH.9	PB2.1	Reserved	PB2.1
DTMnINH.10	Reserved	Reserved	PB2.2
DTMnINH.11	Reserved	Reserved	PB2.3
DTMnINH.12	PB2.4	PB2.4	PB2.4
DTMnINH.13	PB2.5	PB2.5	PB2.5
DTMnINH.14	PB2.6	PB2.6	PB2.6
DTMnINH.15	PB2.7	PB2.7	PB2.7

19.4.3. Interrupts

Within a state structure, the user can selectively enable timeout interrupts and state transition interrupts. The timeout counter and its associated interrupt are enabled using the TOERRIEN flag. If TIOERRIEN is set, TOCOUNT is loaded with the value of TORELOAD on entry into the state. If the TOCOUNT field reaches zero, the TIOERRI interrupt flag will be set, and the state machine transitions to DONE.

The PRISTIEN and SECSTIEN flags enable interrupts upon transition to the primary and secondary states, respectively. When either of these interrupts occurs, the DTMI interrupt flag will be set by hardware.

19.5. Configuring DTM States in Memory

The DTM does not implement a register for each of the 15 user-defined states. The states required for a DTM block must be defined in a block of memory (Flash or RAM), and the DTM is configured to point to the appropriate memory location where the state structures are stored. When a state is entered, the DTM will fetch the structure for that state from the specified memory location and load it into the STATE register.

Each state is a 32-bit, word-aligned value, which should follow the bit pattern defined in the STATE register. States should be organized in a contiguous block of memory from state 0 up to state 14. Once the state structures are configured in memory, the DTM module can be pointed at the base address (the address of state 0) for that memory structure using the STATEADDR register. STATEADDR contains a fully-qualified 32-bit address in memory, where the two LSBs are always cleared to 0 (this forces word-alignment of the address).

The specific address of each state in memory can be calculated as: STATEADDR + (4 x State Number). Figure 19.2 shows how the states should be organized in memory. Note that only the number of states used by the function need be defined. For example, a function requiring only states 0-2 would only need to allocate three 32-bit structures in memory.

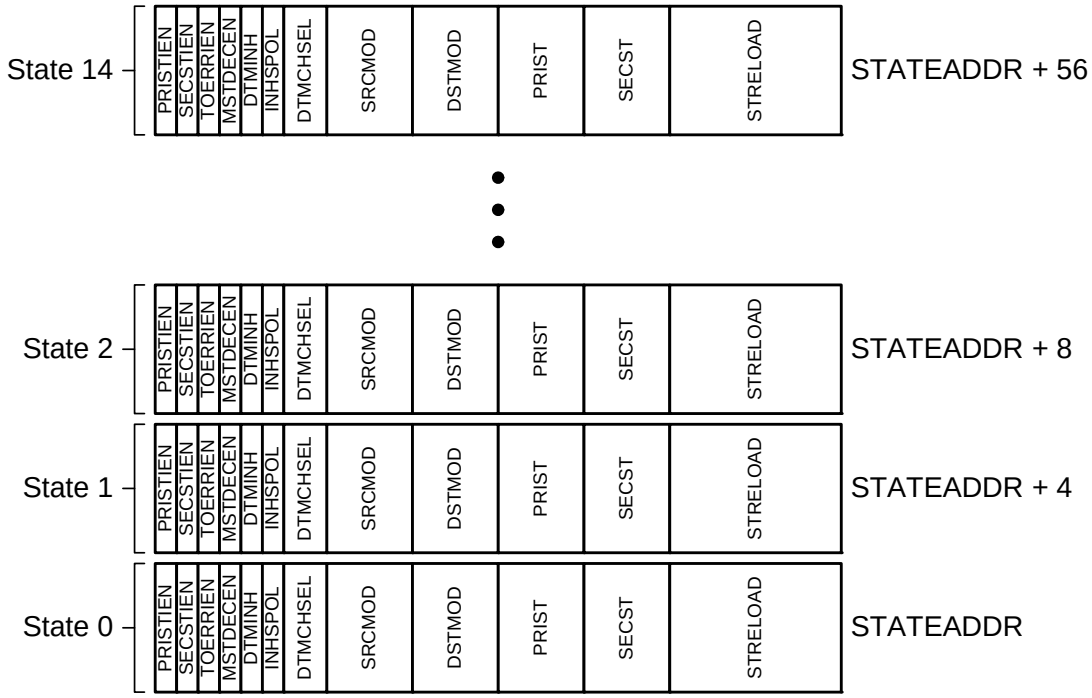


Figure 19.2. State Memory Map

19.6. DTM0, DTM1 and DTM2 Registers

This section contains the detailed register descriptions for DTM0, DTM1 and DTM2 registers.

Register 19.1. DTMn_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTMEN	DTMI	DMAERRI	TOERRI	DTMINH	SRCREQF	DSTREQF	INHF	DBGMD	Reserved			INHSSEL			
Type	RW	RW	RW	RW	RW	R	R	R	RW	R			RW			
Reset	0	0	0	0	0	1	1	X	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LASTST				ST				STCOUNT							
Type	R				RW				RW							
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Register ALL Access Addresses

DTM0_CONTROL = 0x4004_A000
DTM1_CONTROL = 0x4004_B000
DTM2_CONTROL = 0x4004_C000
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 19.4. DTMn_CONTROL Register Bit Descriptions

Bit	Name	Function
31	DTMEN	**Module Enable.** Setting this bit to 1 starts the DTM module at the state value written to the ST field. 0: Disable the DTM module. 1: Enable the DTM module.
30	DTMI	**Module Interrupt Flag.** Hardware sets this bit to 1 and causes a DTM interrupt when the module transitions to states based on the SECSTIEN and PRISTIEN bit settings or a timeout occurs. This bit must be cleared by firmware.
29	DMAERRI	**DMA Error Interrupt Flag.** Hardware sets this bit to 1 and causes a DTM interrupt when a DMA transfer error occurs. This bit must be cleared by firmware.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 19.4. DTMn_CONTROL Register Bit Descriptions

Bit	Name	Function
28	TOERRI	**Timeout Error Interrupt Flag.** Hardware sets this bit to 1 and causes a DTM interrupt when a timeout occurs during a state with timeouts enabled (TOERRIEN = 1). This bit must be cleared by firmware.
27	DTMINH	**DTM Module Inhibit.** Setting this bit inhibits the DTM module. The module will ignore any DMA requests while this bit is set.
26	SRCREQF	**Source Peripheral DMA Request Status Flag.** This flag indicates the DMA request status of the source peripheral indicated by the SRCMOD field.
25	DSTREQF	**Destination Peripheral DMA Request Status Flag.** This flag indicates the DMA request status of the destination peripheral indicated by the DSTMOD field.
24	INHF	**Inhibit Status Flag.** This flag provides the status of the inhibit signal selected by the INHSSEL field.
23	DBGMD	**Debug Mode.** 0: The DTM module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the DTM module to halt.
22:20	Reserved	Must write reset value.
19:16	INHSSEL	**Inhibit Signal Select.** This field selects the DTMnINH.x inhibit signal used by states with the INHSEN bits set.
15:12	LASTST	**Last State.** This read-only field captures the last non-done state of the DTM, allowing firmware to determine which state caused a transition to the current state.
11:8	ST	**Active State.** This field reports the active state of the module. Firmware can write the ST field to set the initial state of the DTM. Hardware sets this field to 15 when the DTM operation completes.
7:0	STCOUNT	**Active State Counter.** This field is an 8-bit state counter for the DTM module that tracks the number of requests generated since the module entered an active state. Hardware automatically updates this field with the STRELOAD value when a state becomes active and decrements this field each time a DMA request is generated. A value of 0 corresponds with 256 requests.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 19.2. DTMn_TIMEOUT: Module Timeout

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TOCOUNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TORELOAD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

DTM0_TIMEOUT = 0x4004_A010
DTM1_TIMEOUT = 0x4004_B010
DTM2_TIMEOUT = 0x4004_C010

Table 19.5. DTMn_TIMEOUT Register Bit Descriptions

Bit	Name	Function
31:16	TOCOUNT	**Timeout Counter.** This field is the current timeout counter value. Hardware automatically reloads this field with TORELOAD when a state becomes active with timeouts enabled (TOERRIEN = 1) and decrements the value to 0.
15:0	TORELOAD	**Timeout Counter Reload.** The timeout period represented by TORELOAD is given by: $T_{TIMEOUT} = 256 \times T_{APB} \times (TORELOAD + 1)$

Register 19.3. DTMn_MSTCOUNT: Master Counter

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	MSTCOUNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

DTM0_MSTCOUNT = 0x4004_A020
DTM1_MSTCOUNT = 0x4004_B020
DTM2_MSTCOUNT = 0x4004_C020

Table 19.6. DTMn_MSTCOUNT Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	MSTCOUNT	**Master Counter.** This field is the total number of DMA operations expected by the DTM module. Firmware must write this field when initializing the DTM module. Hardware decrements the MSTCOUNT field for each DMA request generated by a state with master counter decrementing enabled (MSTDECEN = 1).

Register 19.4. DTMn_STATEADDR: State Address

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	STATEADDR[29:14]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	STATEADDR[13:0]														Reserved	
Type	RW														R	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

DTM0_STATEADDR = 0x4004_A030
DTM1_STATEADDR = 0x4004_B030
DTM2_STATEADDR = 0x4004_C030

Table 19.7. DTMn_STATEADDR Register Bit Descriptions

Bit	Name	Function
31:2	STATEADDR	**State Address.** This field is the word-aligned address to the start of the state configuration values. Hardware generates the effective state configuration word address by adding the current state value (ST) to this field.
1:0	Reserved	Must write reset value.

Register 19.5. DTMn_STATE: Active DTM State

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	PRISTIEN	SECSTIEN	TOERRIEN	MSTDECEN	DTMINH	INHSPOL	DTMCHSEL		SRCMOD				DSTMOD			
Type	R	R	R	R	R	R	R		R				R			
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PRIST				SECST				STRELOAD							
Type	R				R				R							
Reset	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Register ALL Access Addresses

DTM0_STATE = 0x4004_A040
DTM1_STATE = 0x4004_B040
DTM2_STATE = 0x4004_C040

Table 19.8. DTMn_STATE Register Bit Descriptions

Bit	Name	Function
31	PRISTIEN	**Primary State Transition Interrupt Enable.** If this bit is set, hardware will set the DTM interrupt flag (DTMI) and generate an interrupt when this state transitions to its primary state (PRIST).
30	SECSTIEN	**Secondary State Transition Interrupt Enable.** If this bit is set, hardware will set the DTM interrupt flag (DTMI) and generate an interrupt when this state transitions to its secondary state (SECST).
29	TOERRIEN	**Timeout Enable.** If this bit is set, hardware reloads the TOCOUNT counter field with the TORELOAD value when this state becomes active. Hardware will set the TOERRI interrupt flag if this state remains active for longer than the number of APB clocks specified by the TORELOAD field.
28	MSTDECEN	**Master Decrement Enable.** If this bit is set, each DMA request generated by this state will decrement the MSTCOUNT field. If this bit is set and MSTCOUNT reaches 0, the state machine will transition to the secondary state (SECST).

Table 19.8. DTMn_STATE Register Bit Descriptions

Bit	Name	Function
27	DTMINH	**Module Inhibit Enable.** If this bit is set, generated DMA requests will be ignored until the inhibit signal selected by INHSSEL matches the polarity set by INHSPOL. This feature can be is used to stall DMA requests until an external peripheral is ready. If timeouts are enabled (TOERRIEN = 1), the TOCOUNT counter continues to decrement while the inhibit signal is asserted.
26	INHSPOL	**Inhibit Signal Polarity.** This bit selects the polarity of the inhibit signal selected by INHSSEL. 0: A logic low on the pin selected by INHSEL will allow the DTM to proceed. 1: A logic high on the pin selected by INHSEL will allow the DTM to proceed.
25:24	DTMCHSEL	**DTM Channel Select.** 00: Select DTMn channel A for this state. 01: Select DTMn channel B for this state. 10: Select DTMn channel C for this state. 11: Select DTMn channel D for this state.
23:20	SRCMOD	**Source Module.** This field selects the peripheral for the state's source DMA requests. Setting this field to 15 indicates that the state will not require a source DMA request, as the source is RAM, flash or a peripheral that does not generate or require DMA requests.
19:16	DSTMOD	**Destination Module.** This field selects the peripheral for the state's destination DMA requests. Setting this field to 15 indicates that the state will not require a destination DMA request, as the destination is RAM or a peripheral that does not generate or require DMA requests.
15:12	PRIST	**Primary State.** This field sets the primary state of the module.
11:8	SECST	**Secondary State.** This field sets the secondary state of the module.
7:0	STRELOAD	**Active State Counter Reload.** This field sets the reload value for the active state counter (STCOUNT). Hardware automatically updates the STCOUNT field with this value when this state becomes active. A value of 0 corresponds with 256 requests.

19.7. DTMn Register Memory Map

Table 19.9. DTMn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DTMn_CONTROL	0x0	ALL | SET | CLR	DTMEN	DTMI	DMAERRI	TOERRI	DTMINH	SRCREQF	DSTREQF	INHF	DBGMD	Reserved			INHSSEL				LASTST				ST				STCOUNT							
DTMn_TIMEOUT	0x10	ALL	TOCOUNT																TORELOAD															
DTMn_MSTCOUNT	0x20	ALL	Reserved																MSTCOUNT															

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: DTM0 = 0x4004_A000, DTM1 = 0x4004_B000, DTM2 = 0x4004_C000

Table 19.9. DTMn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DTMn_STATEADDR	0x30	ALL	STATEADDR																														Reserved	
DTMn_STATE	0x40	ALL	PRISTIEN	SECSTIEN	TOERRIEN	MSTDECEN	DTMINH	INHSPOL	DTMCHSEL	SRCMOD				DSTMOD				PRIST				SECST			STRELOAD									

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: DTM0 = 0x4004_A000, DTM1 = 0x4004_B000, DTM2 = 0x4004_C000

20. Enhanced Cyclic Redundancy Check (ECRC0)

This section describes the Enhanced Cyclic Redundancy Check (ECRC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the ECRC block, which is used by all device families covered in this document.

20.1. ECRC Features

The ECRC module includes the following features:

- Support for any (user-programmable) 16-bit polynomial, and one fixed 32-bit polynomial.
- Direct peripheral traffic snooping.
- Byte-level bit reversal for the CRC input.
- Byte-order reorientation of words for the CRC input.
- Word or half-word bit reversal of the CRC result.
- Ability to configure and seed an operation in a single register write.
- Support for single-cycle parallel (unrolled) CRC computation for 32- or 8-bit blocks.
- Capability to CRC 32 bits of data per peripheral bus (APB) clock.

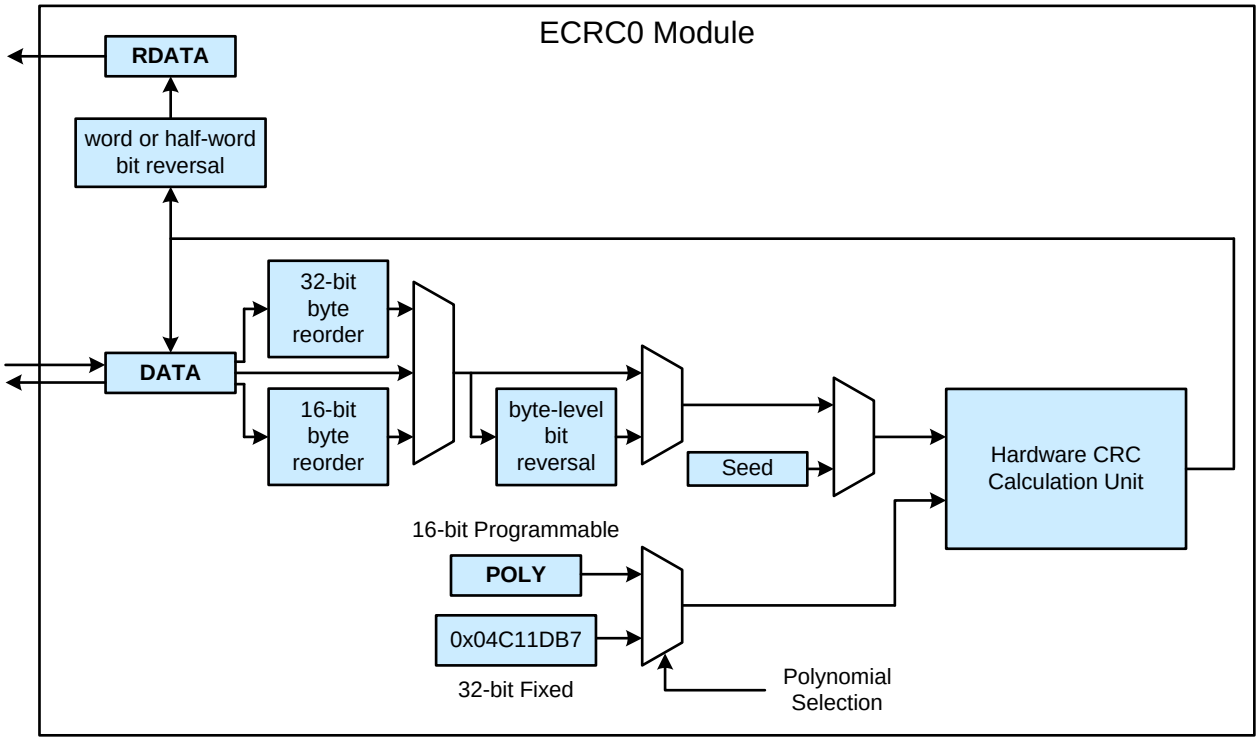


Figure 20.1. ECRC0 Block Diagram

20.2. Overview

The ECRC module is designed to provide hardware calculations for Flash memory verification and communications protocols.

The ECRC module supports 32-bit and 16-bit polynomials. The supported 32-bit polynomial is 0x04C11DB7 (IEEE 802.3). The 16-bit polynomial can be programmed to any value, depending on the needs of the application. Common 16-bit polynomials are 0x1021 (CCITT–16), 0x3D65 (IEC16–MBus), and 0x8005 (ZigBee, 802.15.4, and USB).

The ECRC module will automatically detect non-word writes (byte or half-word) and adjust the internal calculation to only process the least-significant byte or half-word. Any word writes are treated as an entire word and all 32 bits will be considered in the CRC calculation update. The BMDEN bit can be used to force the ECRC module to treat all writes as bytes.

20.3. Polynomial Specification

The POLYSEL bit in the CONTROL register selects between 32-bit and 16-bit polynomial functions. When a 32-bit polynomial is selected, the fixed IEEE 802.3 polynomial (0x04C11DB7) is used. When a 16-bit polynomial is selected, any valid polynomial can be defined by the user in the POLY register.

A valid 16-bit CRC polynomial must have an x^{16} term and an x^0 term. Theoretically, a 16-bit polynomial might have 17 terms total. However, the polynomial SFR is only 16-bits wide. The convention used is to omit the x^{16} term. The polynomial should be written in big endian bit order. The most significant bit corresponds to the highest order term. Thus, the most significant bit in the POLY register represents the x^{15} term, and the least significant bit in the POLY register represents the x^0 term. The least significant bit of POLY cannot be changed, and is always set to 1.

Figure 20.2 depicts the polynomial representation for the CRC-16-CCIT polynomial $x^{16} + x^{12} + x^5 + 1$, or 0x1021.

POLY = 0x1021

	POLY															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

$x^{16}+$ $x^{12}+$ x^5+ 1

Figure 20.2. Polynomial Representation

20.4. Automatic Seeding

The ECRC block can be automatically seeded using the SINTEN and SEED bits in the CONTROL register. When SINTEN is written to 1, the seed value determined by the SEED bit will be seeded to the CRC block. When SEED is 0, the seed is all 0's, and when SEED is 1, the SEED is all 1's.

20.5. Peripheral Data Snooping

The ECRC module has the capability to monitor APB bus traffic to and from certain peripheral blocks, and automatically calculate the CRC on that data. Ten different peripheral blocks can be selected to monitor with the ECRC engine. Within each peripheral block, any word address can be monitored for writes or reads on the APB bus. The SCONTROL register is used to set up the ECRC peripheral for bus snooping. The field SPERISEL is used to select which peripheral block is to be monitored. Table 20.1 shows the different peripheral options available. Note that the Port I/O section encompasses all three of the port I/O control types, and the offset address is the 4kB memory region that contains all of these registers. The SADDR field is used to specify the memory offset within the peripheral block to monitor. SADDR represents bits 11:2 of the memory address, and thus the ECRC snoop function will operate only on word-aligned memory boundaries within each peripheral region. Finally, the SDIRSEL bit defines whether the ECRC module will monitor reads from or writes to the specified memory location.

Table 20.1. ECRC0 Peripheral Data Snooping Selection

SPERISEL Setting	Peripheral to Monitor	Peripheral Offset Address
0000	USART0	0x4000_0000
0001	UART0	0x4000_1000
0010	SPI0	0x4000_4000
0011	SPI1	0x4000_5000
0100	I2C0	0x4000_9000
0101	SARADC0	0x4001_A000
0110	AES0	0x4002_7000
0111	ENCDEC0	0x4004_F000
1000	Port I/O (PBCFG, PBSTD and PBGP)	0x4002_A000
1001	FLASHCTRL0	0x4002_E000

For example, to configure the ECRC module to monitor data sent to UART0, the SPERISEL field should be set to 0001, the SADDR field should be set to 0x1C, and the SDIRSEL bit should be cleared to 0. This will monitor any APB writes to the UART0_DATA register at address 0x4000_1070. The address is calculated as the UART0 base address (0x4000_1000) plus the SADDR field left-shifted by two bits (0x070).

Peripheral data snooping should be configured prior to setting up the rest of the CRC operation. Once the peripheral has been configured as desired, the snooping function is enabled by setting SEN in the SCONTROL register to 1. Note that the ECRC block does not initiate a read or write of the selected peripheral. Rather, the snooping feature is monitoring the APB bus for traffic, and will grab the data when a bus master such as the core or the DMA engine reads the specified address.

20.6. DMA Configuration and Usage

A DMA channel may be used to transfer data into the ECRC module. The ECRC module supports byte, half-word, and word writes. All byte and half-word writes must be word-aligned. The recommended DMA usage model is to use the DMA to transfer all available words of data and use firmware writes to capture any remaining bytes. To write data into the ECRC module, the DMA must move one word of data at a time from the source location in memory to the internal DATA register in non-incrementing mode. Firmware can then write any remaining bytes to the DATA register and read the CRC result from the DATA register. The ECRC DATA register should not be directly written to or read from when targeted by a DMA channel.

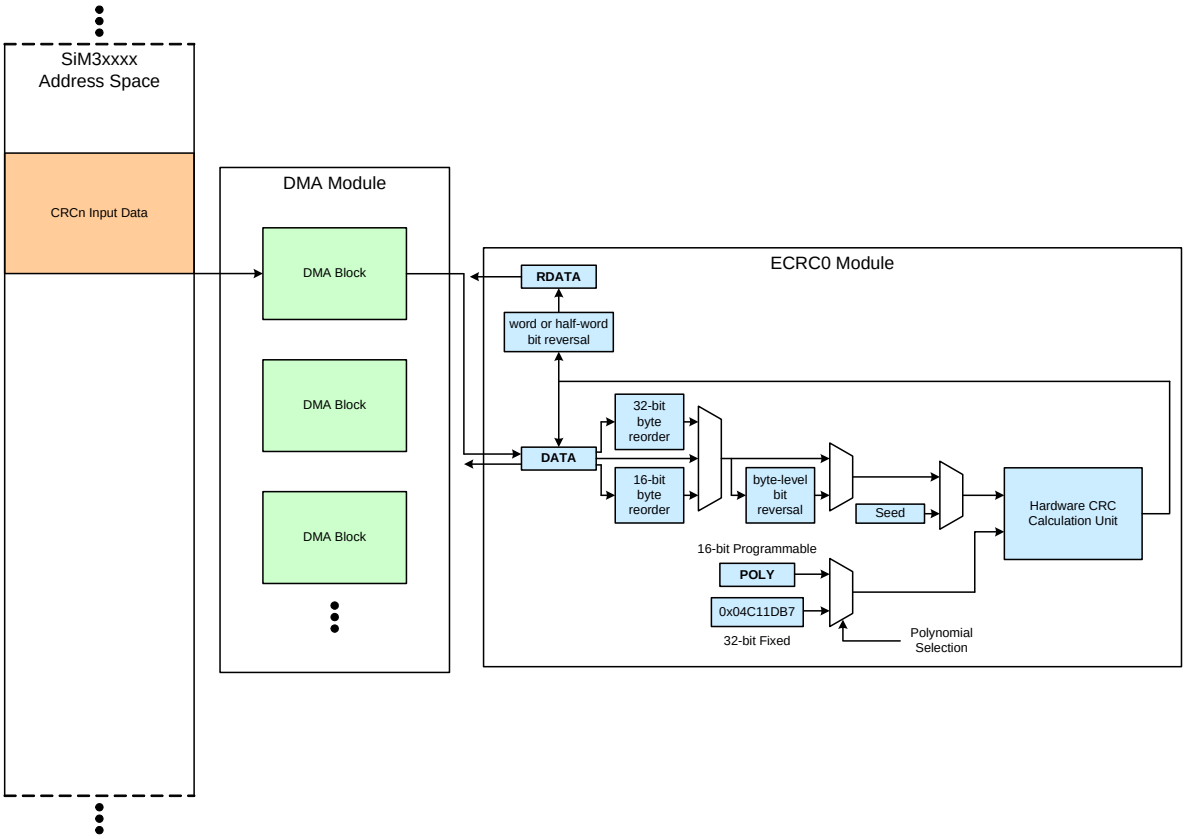


Figure 20.3. ECRC DMA Configuration

For the ECRC module, the DMA channel should be set up as follows:

- Source size and Destination Size are 2 for a word transfer.
- Number of transfers is N - 1, where N is the number of 4-byte words.
- RPOWER = 0 (1 word transfer per transaction).

To start a DMA operation with the ECRC module out of any device reset:

1. Set up the DMA channel for the CRC Input.
2. Configure the ECRC peripheral operation in the CONTROL register.
3. Start the DMA transfer.
4. Wait for DMA completion interrupt.
5. Write any remaining bytes to the DATA register to complete the CRC calculation for the memory block.
6. Read the CRC result from DATA or the bit-reversed CRC result from RDATA.

20.7. Byte-Level Bit Reversal and Byte Reordering

The byte-level bit reversal and byte reordering operations occur before the data is used in the CRC calculation. Byte reordering can occur on words or half words. The hardware ignores the ORDER field with any byte writes or operations with byte mode enabled (BMDEN = 1), but the bit reversal settings (BBREN) are still applied to the byte.

Big-endian data can be treated like 32-bit little endian data, as shown in Figure 20.4. In this example, ORDER is set to 10b for big-endian byte ordering, and BBREN is set to 1 for byte-level bit reversal.

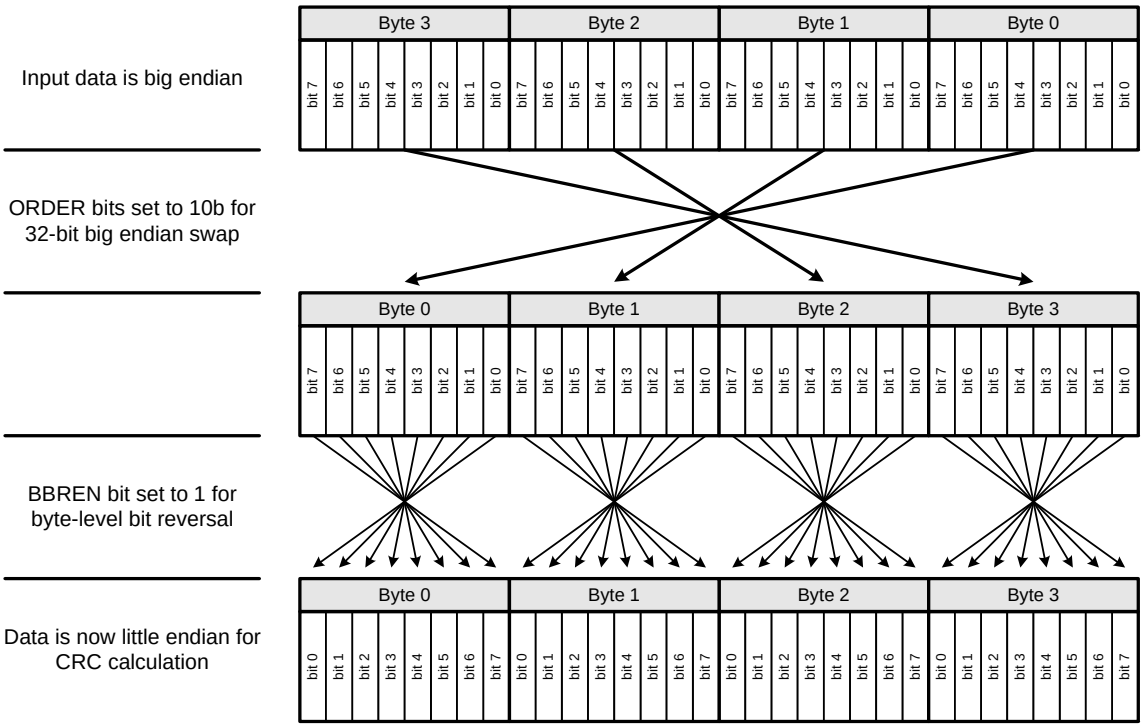


Figure 20.4. ECRC Data Ordering Example—Big Endian to Little Endian

Big-endian data can be treated like 16-bit little endian data with MSB-first bit ordering, as shown in Figure 20.5. In this example, ORDER is set to 01b for 16-bit big-endian byte order, and BBREN is set to 1 for byte-level bit reversal.

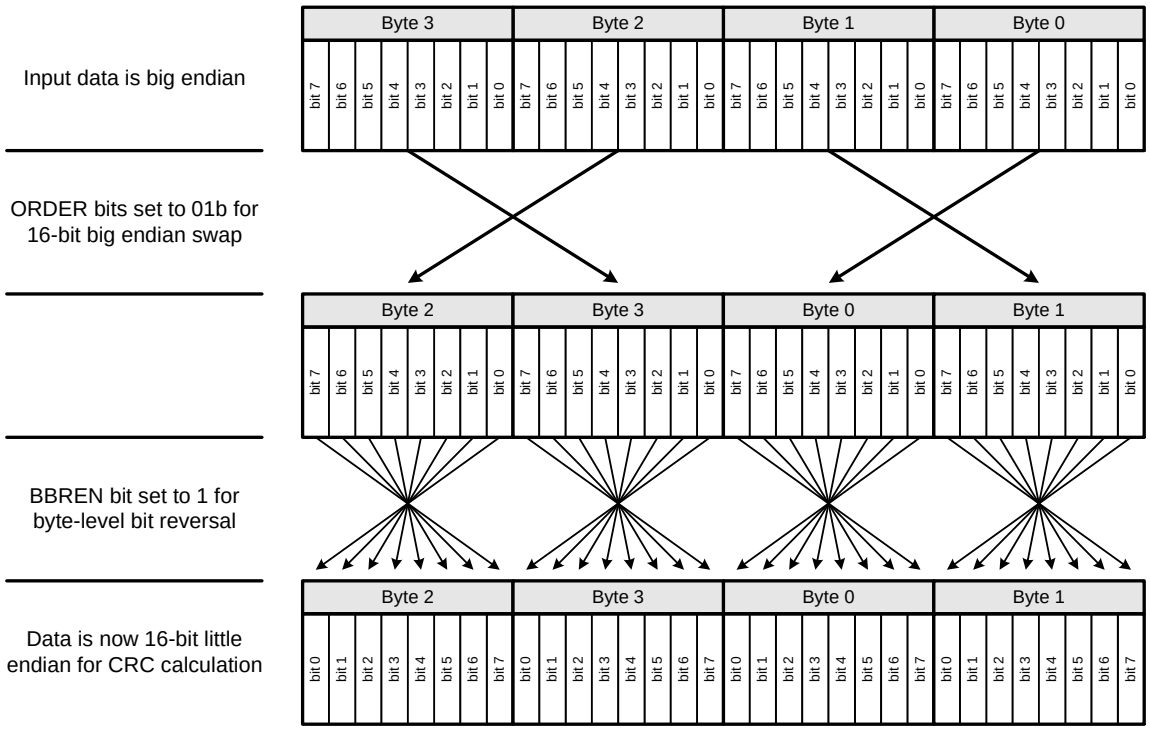


Figure 20.5. ECRC Data Ordering Example—Big Endian to 16-bit Little Endian

Assuming a word input byte order of B3 B2 B1 B0, the values used in the CRC calculation for the various settings of the byte-level bit reversal and byte reordering are shown in Table 20.2.

Table 20.2. Byte-Level Bit Reversal and Byte Reordering Results (B3 B2 B1 B0 Input Order)

Original CRC Calculation Method			Equivalent ECRC Settings		
Polynomial Width (bits)	**Byte Order**	**Bit Order (MSB/LSB first)**	**ORDER bits setting**	**BBREN bit setting**	**Input to CRC calculation**
32	little endian	LSB	00 (none)	0	B3 B2 B1 B0
32	little endian	MSB	10 (32-bit)	1	‘B0 ‘B1 ‘B2 ‘B3
32	big endian	LSB	10 (32-bit)	0	B0 B1 B2 B3
32	big endian	MSB	00 (none)	1	‘B3 ‘B2 ‘B1 ‘B0
16	little endian	LSB	00 (none)	0	B3 B2 B1 B0
16	little endian	MSB	01 (16-bit)	1	‘B2 ‘B3 ‘B0 ‘B1
16	big endian	LSB	01 (16-bit)	0	B2 B3 B0 B1
16	big endian	MSB	00 (none)	1	‘B3 ‘B2 ‘B1 ‘B0
8	—	LSB	—	0	XX XX XX B0
8	—	MSB	—	1	XX XX XX ‘B0

Notes:
1. X indicates a “don’t care.”
2. Bn is the byte field within the word.
3. ‘Bn is the bit-reversed byte field within the word.

20.8. ECRC0 Registers

This section contains the detailed register descriptions for ECRC0 registers.

Register 20.1. ECRC0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	ASEEDSEL	ASEEDEN	Reserved	ORDER		BBREN	BMDEN	Reserved			POLYSEL	Reserved	CRCEN	SEED	SINITEN
Type	R	RW	RW	R	RW		RW	RW	R			RW	R	RW	RW	W
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Register ALL Access Address

ECRC0_CONTROL = 0x4002_8000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 20.3. ECRC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:15	Reserved	Must write reset value.
14	ASEEDSEL	**Automatic Seed Byte Select.** This bit selects the byte of DATA, RDATA, or BRDATA that causes a re-seed of the CRC result when automatic seeding is enabled. The selected byte can be read with word, half-word or byte read operations. 0: Select a read of the least-significant byte ([7:0]) of DATA, RDATA or BRDATA for automatic re-seeding. 1: Select a read of the most-significant byte [31:24] for 32-bit operations, and [15:8] for 16-bit operations) of DATA, RDATA or BRDATA for automatic re-seeding.
13	ASEEDEN	**Automatic Seed Enable.** 0: Disable automatic seeding. 1: Enable automatic seeding. Reading the byte of the DATA register selected by ASEEDSEL re-seeds the CRC result with the setting selected by SEED.
12	Reserved	Must write reset value.

Table 20.3. ECRC0_CONTROL Register Bit Descriptions

Bit	Name	Function
11:10	ORDER	**Input Processing Order.** Controls the input order of the bytes in a half-word or word to the CRC calculation unit. This setting has no effect in byte mode. 00: No byte reorientation (output is same order as input). 01: Swap for 16-bit big endian order (input: B3 B2 B1 B0, output: B2 B3 B0 B1). 10: Swap for 32-bit big endian order (input: B3 B2 B1 B0, output: B0 B1 B2 B3). 11: Reserved.
9	BBREN	**Byte-Level Bit Reversal Enable.** Controls the input order of the bits in each byte to the CRC calculation unit. When set to 1, byte-level bit reversal is enabled and the bits in each byte are reversed. 0: No byte-level bit reversal (input is same order as written). 1: Byte-level bit reversal enabled (the bits in each byte are reversed).
8	BMDEN	**Byte Mode Enable.** Enables byte mode, where all writes to the DATA register are considered as 8-bit writes. When set to 1, only the least-significant byte of the data word will be used for the CRC calculation. 0: Disable byte mode (word/byte width is determined automatically by the hardware). 1: Enable byte mode (all writes are considered as bytes).
7:5	Reserved	Must write reset value.
4	POLYSEL	**Polynomial Selection.** Selects the polynomial used by the CRC calculations. 0: Select the fixed 32-bit polynomial: 0x04C11DB7. 1: Select the programmable 16-bit polynomial. The POLY register sets the polynomial coefficients.
3	Reserved	Must write reset value.
2	CRCEN	**CRC Enable.** Enables CRC functionality. Until this bit is set to 1, writes to the DATA register move into the CRC results register without any CRC operation. Byte reorientation and bit reversal is still active even if CRCEN is cleared to 0. 0: Disable CRC operations. 1: Enable CRC operations.
1	SEED	**Seed Setting.** Determines the value of all CRC register bits when a 1 is written to the SINITEN bit. This bit always reads back as 0. The desired value of SEED must be written in the same register write that sets SINITEN. 0: CRC seed value is all 0's (0x00000000) 1: CRC seed value is all 1's (0xFFFFFFFF).

Table 20.3. ECRC0_CONTROL Register Bit Descriptions

Bit	Name	Function
0	SINITEN	**Seed Initialization Enable.** When this bit is set to 1, all bits of the CRC register are initialized to the value written to the SEED bit. This bit always reads back as 0. SINITEN and SEED should be written in the same register write. 0: Do not initialize the CRC module to the value set by the SEED bit. 1: Initialize the CRC module to the value set by the SEED bit.

Register 20.2. ECRC0_POLY: 16-bit Programmable Polynomial

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	POLY															
Type	RW															
Reset	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1

Register ALL Access Address

ECRC0_POLY = 0x4002_8010

Table 20.4. ECRC0_POLY Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	POLY	**16-bit Programmable Polynomial.** This field is the 16-bit programmable CRC polynomial. A 1 in an N bit position adds an x^N term to the CRC polynomial. The least-significant bit of this field is read-only and always reads back 1.

Register 20.3. ECRC0_DATA: Input/Result Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ECRC0_DATA = 0x4002_8020

Table 20.5. ECRC0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**Input/Result Data.** If the CRCEN bit is set to 1, data written to this register will be used for CRC calculations, byte-level bit reversal, and byte reordering. The current CRC result can be read from this register at any time. If the CRCEN bit is cleared to 0, data written to this register will be used for byte-level bit reversal and byte reordering only. No delay is required between writing the data and reading the CRC or reordering result.

Notes:

1. When reading this register with ASEEDEN = 1, byte or half-word access should read the portion specified by ASEEDSEL last.

Register 20.4. ECRC0_RDATA: Bit-Reversed Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RDATA[31:16]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RDATA[15:0]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ECRC0_RDATA = 0x4002_8030

Table 20.6. ECRC0_RDATA Register Bit Descriptions

Bit	Name	Function
31:0	RDATA	**Bit-Reversed Output Data.** This register provides the bit reversed version of the DATA register. When the 32-bit CRC polynomial is selected, the reversal occurs on the entire 32-bit word. When a 16-bit CRC polynomial is selected, the least-significant half-word (bits [15:0]) is reversed.

Notes:

1. When reading this register with ASEEDEN = 1, byte or half-word access should read the portion specified by ASEEDSEL last.

Register 20.5. ECRC0_BRDATA: Byte-Reversed Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BRDATA[31:16]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BRDATA[15:0]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ECRC0_BRDATA = 0x4002_8040

Table 20.7. ECRC0_BRDATA Register Bit Descriptions

Bit	Name	Function
31:0	BRDATA	**Byte-Reversed Output Data.** This register provides the byte-reversed version of the DATA register. When a 32-bit CRC polynomial is selected, the bytes reorder to [B0, B1, B2, B3]. When a 16-bit CRC polynomial is selected, the bytes reorder to [B2, B3, B0, B1].

Register 20.6. ECRC0_SCONTROL: Bus Snooping Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				SADDR										Reserved	
Type	R				RW										R	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								SPERISEL				Reserved		SDIRSEL	SEN
Type	R								RW				R		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ECRC0_SCONTROL = 0x4002_8050

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 20.8. ECRC0_SCONTROL Register Bit Descriptions

Bit	Name	Function
31:28	Reserved	Must write reset value.
27:18	SADDR	**Snooping Address.** This field sets the word base address of the selected device peripheral for the module to snoop. This value corresponds to APB address bits [11:2] and selects word addresses.
17:8	Reserved	Must write reset value.
7:4	SPERISEL	**Snooping Peripheral Select.** Setting this field to x selects the ECRCn.x device peripheral as the module to snoop.
3:2	Reserved	Must write reset value.
1	SDIRSEL	**Snooping Direction Select.** 0: ECRC will snoop writes to the selected peripheral. 1: ECRC will snoop reads from the selected peripheral.
0	SEN	**Snooping Enable.** When enabled, the ECRC module will snoop the APB bus accesses for the selected peripheral and automatically CRC the data on the bus. To avoid incorrect CRC calculations, writes to DATA should not be performed when snooping the bus. 0: Disable automatic bus snooping. 1: Enable automatic bus snooping.

20.9. ECRC0 Register Memory Map

Table 20.9. ECRC0 Memory Map

Register Name	ECRC0_CONTROL	ECRC0_POLY	ECRC0_DATA	ECRC0_RDATA
ALL Address	0x4002_8000	0x4002_8010	0x4002_8020	0x4002_8030
Access Methods	ALL | SET | CLR	ALL	ALL	ALL
Bit 31	Reserved	Reserved	DATA	RDATA
Bit 30				
Bit 29				
Bit 28				
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23				
Bit 22				
Bit 21				
Bit 20				
Bit 19				
Bit 18				
Bit 17				
Bit 16				
Bit 15		POLY		
Bit 14	ASEEDSEL			
Bit 13	ASEEDEN			
Bit 12	Reserved			
Bit 11	ORDER			
Bit 10				
Bit 9	BBREN			
Bit 8	BMDEN			
Bit 7	Reserved			
Bit 6				
Bit 5				
Bit 4	POLYSEL			
Bit 3	Reserved			
Bit 2	CRCEN			
Bit 1	SEED			
Bit 0	SINITEN			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 20.9. ECRC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ECRC0_BRDATA	0x4002_8040	ALL	BRDATA																															
ECRC0_SCONTROL	0x4002_8050	ALL | SET | CLR	Reserved				SADDR											Reserved								SPERISEL				Reserved			SDIRSEL	SEN

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

21. Encoder / Decoder (ENCDEC0)

This section describes the Encoder/Decoder (ENCDEC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the ENCDEC block, which is used by all device families covered in this document.

21.1. ENCDEC Features

The ENCDEC module includes the following features:

- Supports Manchester and Three-out-of-Six encoding and decoding.
- Automatic flag clearing when writing the input or reading the output data registers.
- Writing to the input data register automatically initiates an encode or decode operation.
- Optional output in one’s complement format.
- Hardware error detection for invalid input data during decode operations.
- Flexible byte swapping on the input or output data.

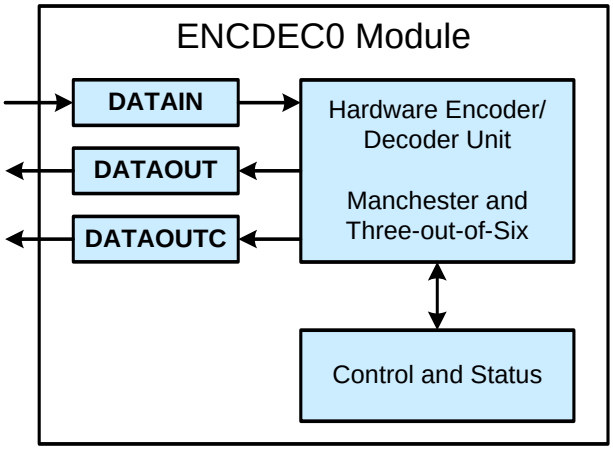


Figure 21.1. ENCDEC0 Block Diagram

21.2. Manchester Encoding

In Manchester encoding, each bit is encoded as two bits. To encode Manchester data, the OPMD bit should be cleared to 0 (for Manchester), and the EDMD bit should be set to 1 (encode).

Data to encode is written to the DATAIN register. DATAIN allows byte, half-word and word access (word-aligned only). Input is processed one byte at a time according to the order established by the IORDER field, so 1, 2, or 4 bytes may be processed in a single write to DATAIN. When all of the data written to DATAIN has been processed, the INRDYI flag in the STATUS register is set to indicate that additional data can be written and a DMA request can be asserted.

Each output in Manchester encoding mode is two bytes wide. When encoding data, the user can select whether to proceed with one or two encode operations using the MOSIZE bit in the CONTROL register. When the desired number of encode operations have been completed, hardware will set the ORDYI flag in the STATUS register. If additional data is pending in the DATAIN register, the data encoding will continue (and ORDYI will automatically be cleared) when the DATAOUT or DATAOUTC registers are read. All available data must be read from DATAOUT or DATAOUTC with a single read operation. Table 21.1 shows the output of the Manchester encode operation.

Table 21.1. Manchester Encoding

Input Data			Encoded Output		
nibble			byte		
dec	hex	bin	bin	hex	dec
0	0	0000	10101010	AA	170
1	1	0001	10101001	A9	169
2	2	0010	10100110	A6	166
3	3	0011	10100101	A5	165
4	4	0100	10011010	9A	154
5	5	0101	10011001	99	153
6	6	0110	10010110	96	150
7	7	0111	10010101	95	149
8	8	1000	01101010	6A	106
9	9	1001	01101001	69	105
10	A	1010	01100110	66	102
11	B	1011	01100101	65	101
12	C	1100	01011010	5A	90
13	D	1101	01011001	59	89
14	E	1110	01010110	56	86
15	F	1111	01010101	55	85

21.3. Manchester Decoding

Decoding manchester data is symmetrical to the encode operation. Two bits are decoded to a single bit. To decode Manchester data, the OPMD bit should be cleared to 0 (for Manchester), and the EDMD bit should be cleared to 0 (decode).

Data to decode is written to the DATAIN register. DATAIN access should be restricted to half-word and word writes (word-aligned only) in this mode. The DERRI flag will be set if an incompatible register access is attempted. Input is processed two bytes at a time according to the order established by the IORDER field, so a half-word write will produce one output, and a word write will produce two outputs. When all of the data written to DATAIN has been processed, the INRDYI flag in the STATUS register is set to indicate that additional data can be written.

Each output in Manchester decoding mode is one byte wide. When decoding data, the user can select whether to proceed with one or two decode operations using the MOSIZE bit in the CONTROL register. When the desired number of decode operations have been completed, hardware will set the ORDYI flag in the STATUS register. If additional data is pending in the DATAIN register, the data encoding will continue (and ORDYI will automatically be cleared) when the DATAOUT or DATAOUTC registers are read. All available data must be read from DATAOUT or DATAOUTC with a single read operation. Table 21.2 shows the output of the Manchester decode operation. The DERRI flag will be set if an invalid input is written.

Table 21.2. Manchester Decoding

Input			Decoded Output		
Byte			Nibble		
bin	hex	dec	dec	hex	bin
01010101	55	85	15	F	1111
01010110	56	86	14	E	1110
01011001	59	89	13	D	1101
01011010	5A	90	12	C	1100
01100101	65	101	11	B	1011
01100110	66	102	10	A	1010
01101001	69	105	9	9	1001
01101010	6A	106	8	8	1000
10010101	95	149	7	7	0111
10010110	96	150	6	6	0110
10011001	99	153	5	5	0101
10011010	9A	154	4	4	0100
10100101	A5	165	3	3	0011
10100110	A6	166	2	2	0010
10101001	A9	169	1	1	0001
10101010	AA	170	0	0	0000

21.4. Three-out-of-Six Encoding

Three out of six encoding is similar to Manchester encoding. In Three-out-of-Six encoding each nibble is encoded as a six-bit symbol. Four nibbles (two bytes) are encoded as 24-bits (three bytes).

To use Three-out-of-Six encoding, the OPMD bit should be set to 1 (for Three-out-of-Six), and the EDMD bit should be set to 1 (encode).

Data to encode is written to the DATAIN register. DATAIN access should be restricted to half-word and word writes (word-aligned only) in this mode. The DERRI flag will be set if an incompatible register access is attempted. Input is processed two bytes at a time according to the order established by the IORDER field, so a half-word write will produce one output, and a word write will produce two outputs. When all of the data written to DATAIN has been processed, the INRDYI flag in the STATUS register is set to indicate that additional data can be written.

Each output in Three-out-of-Six encoding mode is three bytes wide. When each encode operation is complete, hardware will set the ORDYI flag in the STATUS register. If additional data is pending in the DATAIN register, the data encoding will continue (and ORDYI will automatically be cleared) when the DATAOUT or DATAOUTC registers are read. All available data must be read from DATAOUT or DATAOUTC with a single read operation. Table 21.3 shows the output of the Three-out-of-Six encode operation.

Table 21.3. Three-out-of-Six Encoding Nibble

Input			**Encoded Output**			
nibble			**symbol**			
dec	hex	bin	bin	dec	hex	octal
0	0	0000	010110	22	16	26
1	1	0001	001101	13	0D	15
2	2	0010	001110	14	0E	16
3	3	0011	001011	11	0B	13
4	4	0100	011100	28	1C	34
5	5	0101	011001	25	19	31
6	6	0110	011010	26	1A	32
7	7	0111	010011	19	13	23
8	8	1000	101100	44	2C	54
9	9	1001	100101	37	25	45
10	A	1010	100110	38	26	46
11	B	1011	100011	35	23	43
12	C	1100	110100	52	34	64
13	D	1101	110001	49	31	61
14	E	1110	110010	50	32	62
15	F	1111	101001	41	29	51

21.5. Three-out-of-Six Decoding

Three-out-of-Six decoding is a similar inverse process. In Three-out-of-Six decoding, every six-bit symbol decodes to a nibble.

To decode Three-out-of-Six data, the OPMD bit should be set to 1 (for Three-out-of-Six), and the EDMD bit should be cleared to 0 (decode).

Data to decode is written to the DATAIN register. DATAIN access should be restricted to word writes only in this mode. The DERRI flag will be set if an incompatible register access is attempted. Input is processed three bytes at a time according to the order established by the IORDER field, so a word write will produce a single output. The most significant byte of the input word is not used. When the data written to DATAIN has been processed, the INRDYI flag in the STATUS register is set to indicate that additional data can be written.

Each output in Three-out-of-Six decoding mode is two bytes wide. When an encode operation is complete, hardware will set the ORDYI flag in the STATUS register. The ORDYI flag will automatically be cleared when the DATAOUT or DATAOUTC registers are read. All available data must be read from DATAOUT or DATAOUTC with a single read operation. Table 21.4 shows the output of the Three-out-of-Six decode operation. The DERRI flag will be set if an invalid input is written.

Table 21.4. Three-out-of-Six Decoding

Input			**Decoded Output**		
Symbol			**Nibble**		
bin	octal	dec	dec	hex	bin
001011	13	11	3	3	0011
001101	15	13	1	1	0001
001110	16	14	2	2	0010
010011	23	19	7	7	0111
010110	26	22	0	0	0000
011001	31	25	5	5	0101
011010	32	26	6	6	0110
011100	34	28	4	4	0100
100011	43	35	11	B	1011
100101	45	37	9	9	1001
100110	46	38	10	A	1010
101001	51	41	15	F	1111
101100	54	44	8	8	1000
110001	61	49	13	D	1101
110010	62	50	14	E	1110
110100	64	52	12	C	1100

21.6. Interrupts and Error Conditions

The Encoder/Decoder peripheral has several interrupt sources available to the user. These interrupts are all indicated in the STATUS register, while enable bits to allow the status flag to trigger an interrupt are located in the CONTROL register. Interrupts may be triggered on the normal data input and output status flags, or when an error condition occurs.

Table 21.5. Interrupt Flags and Conditions

Interrupt Flag (STATUS register)	Enable Bit (CONTROL register)	Conditions
INRDYI	INRDYIEN	Set when new data can be accepted. Cleared by writing to DATAIN (direct or via DMA).
ORDYI	ORDYIEN	Set when output data is ready. Cleared by reading DATAOUT or DATAOUTC (direct or via DMA)
DERRI	DERRIEN	Set when a data input format error occurs. Cleared by firmware.
DURI	DERRIEN	Set when DATAIN is written before new data can be accepted. Cleared by firmware.
DORI	DERRIEN	Set when DATAOUT or DATAOUTC is read, but no new data is available. Cleared by firmware.

Normally, when used with the DMA, the DMA will transfer the entire specified transfer size to and from the Encoder/Decoder peripheral. If a decoder error occurs, decoding will continue until all data has been decoded. The error bit in the DERRI flag will indicate if an error has occurred anywhere in the DMA transfer. Some applications will discard the entire packet after a single decoder error. Aborting the decoder operation at the first decoder error will conserve energy and minimize packet receiver turn-around time. The decoder interrupt service routine should disable the DMA channels associated with the ENCDEC0 peripheral and clear any pending interrupt flags when this occurs.

21.7. DMA Configuration and Usage

A DMA channel may be used to transfer data to and from the ENCDEC module. The DMA's ENCDEC0 TX channel is the input channel, which is used to transfer data into the encoder/decoder via the DATAIN register. Likewise, the ENCDEC0 RX channel of the DMA is the output channel, and is used to transfer data out of the encoder/decoder via the DATAOUT (or DATAOUTC) register. Depending on the encoding scheme and whether the data is being encoded or decoded, the input and output of the encoder/decoder can accept word, half-word or byte transfers.

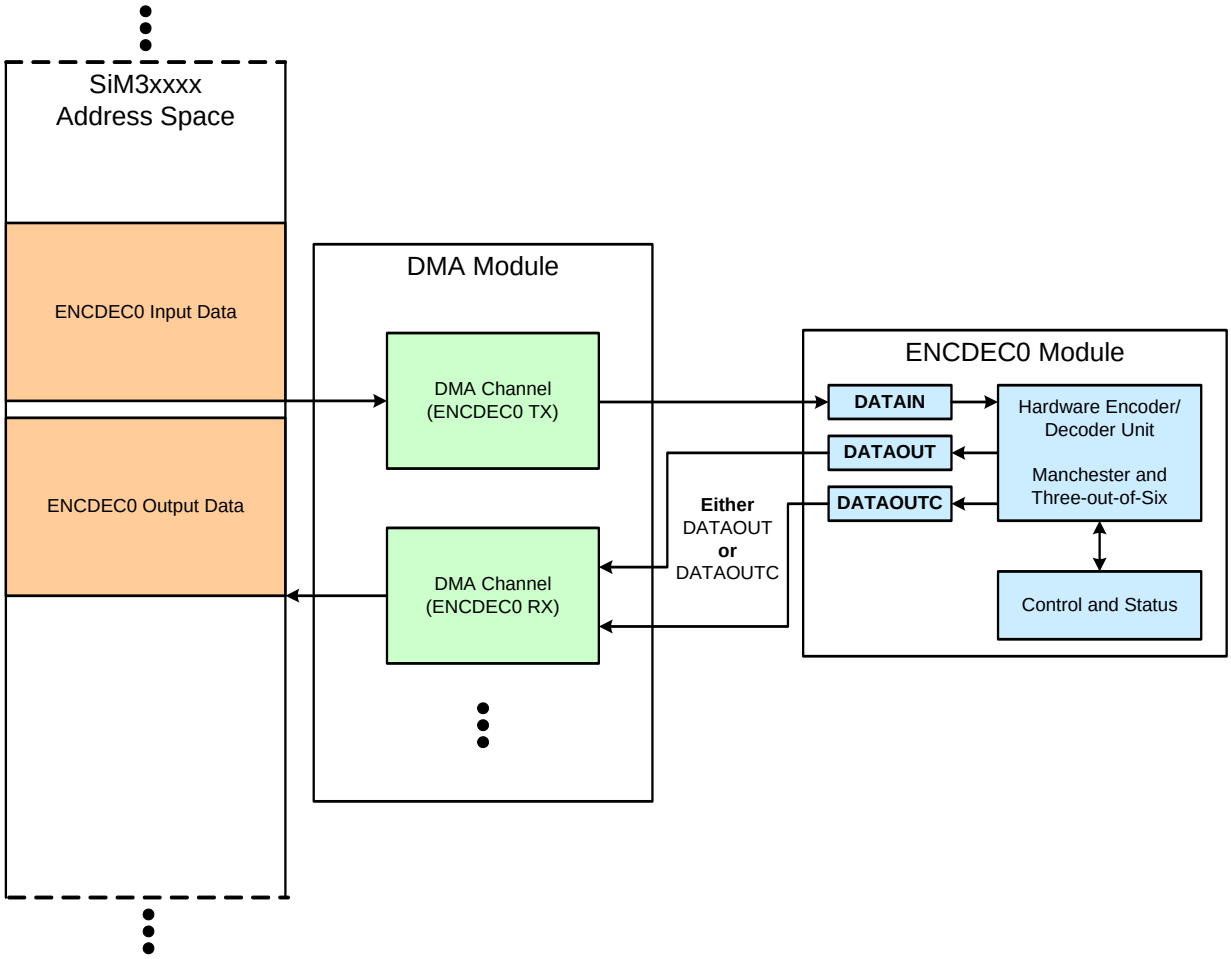


Figure 21.2. ENCDEC0 DMA Configuration

The DMA channel configuration for the ENCDEC block is as follows:

- For simplicity, the input and output channels can be configured for the same number of transfers, but they need not be. Table 21.6 reflects only configurations where the number of input and output transfers are equal. The RPOWER field should be set to 0.
- The ENCDEC0 TX channel must be configured with the data destination set to the DATAIN register in non-incrementing mode (DSTAIMD = 3).
- The ENCDEC0 RX channel must be configured with the data source set to the DATAOUT or DATAOUTC register in non-incrementing mode (SRCAIMD = 3).
- For transfers to/from a memory buffer, Table 21.6 summarizes the recommended settings for the DMA transfer size and increment fields.

Table 21.6. DMA Channel Size and Increment Parameters

Operation	**ENCDEC0 TX Channel** DSTSIZE, SRCAIMD, and SRCSIZE	**ENCDEC0 RX Channel** DSTAIMD, DSTSIZE, and SRCSIZE
Manchester encode, MOSIZE = 0 (small)	0 (byte)	1 (half-word)
Manchester decode, MOSIZE = 0 (small)	1 (half-word)	0 (byte)
Manchester encode, MOSIZE = 1 (large)	1 (half-word)	2 (word)
Manchester decode, MOSIZE = 1 (large)	2 (word)	1 (half-word)
Three-out-of-Six encode	1 (half-word)	2 (word)
Three-out-of-Six decode	2 (word)	1 (half-word)

To start a DMA operation with the ENCDEC module:

1. Configure the ENCDEC module for the desired encode/decode operation.
2. Reset the ENCDEC module using the RESET bit, and clear the DATAOUT register to 0.
3. Configure the ENCDEC0 RX and ENCDEC0 TX DMA channels for the desired encode/decode operation.
4. Enable the DMA channels.
5. Enable DMA mode in the ENCDEC block by setting the DMAEN bit to 1.

21.8. ENCDEC0 Registers

This section contains the detailed register descriptions for ENCDEC0 registers.

Register 21.1. ENCDEC0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	IORDER		OORDER		Reserved	DBGMD	DMAEN	BEN	Reserved	OPMD	EDMD	MOSIZE	RESET	ERRIEN	ORDYIEN	INRDYIEN
Type	RW		RW		R	RW	RW	RW	R	RW	RW	RW	W	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ENCDEC0_CONTROL = 0x4004_F000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 21.7. ENCDEC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:14	IORDER	**Input Order Mode.** 00: Data written to DATAIN is processed in the order written (input: B3 B2 B1 B0, output: B3 B2 B1 B0). 01: The module flips the DATAIN input data in half-words (input: B2 B3 B0 B1, output: B3 B2 B1 B0). 10: The module flips the DATAIN input data in words (input: B0 B1 B2 B3, output: B3 B2 B1 B0). 11: The module flips the lower three bytes of the DATAIN input data (input: B3 B0 B1 B2, output: B3 B2 B1 B0).
13:12	OORDER	**Output Order Mode.** 00: The module outputs data to DATAOUT in the same order as it was processed (input: B3 B2 B1 B0, output: B3 B2 B1 B0). 01: The module flips the data in half-words before outputting to DATAOUT (input: B3 B2 B1 B0, output: B2 B3 B0 B1). 10: The module flips the data in words before outputting to DATAOUT (input: B3 B2 B1 B0, output: B0 B1 B2 B3). 11: The module flips the lower three bytes before outputting to DATAOUT (input: B3 B2 B1 B0, output: B3 B0 B1 B2).
11	Reserved	Must write reset value.

Table 21.7. ENCDEC0_CONTROL Register Bit Descriptions

Bit	Name	Function
10	DBGMD	**Debug Mode.** 0: The ENCDEC module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the ENCDEC module to halt.
9	DMAEN	**DMA Mode Enable.** 0: Disable DMA mode. 1: Enable DMA mode.
8	BEN	**Bypass Encoder/Decoder Operation Enable.** If this bit is set to 1, the ENCDEC module hardware is bypassed, which allows firmware to use the module as a memory copy. 0: Do not bypass ENCDEC operations. 1: Bypass ENCDEC operations.
7	Reserved	Must write reset value.
6	OPMD	**Operation Mode.** 0: The operation selected by EDMD uses Manchester mode. 1: The operation selected by EDMD uses Three-out-of-Six mode.
5	EDMD	**Encode Decode Mode.** 0: Decode data written to DATAIN. 1: Encode data written to DATAIN.
4	MOSIZE	**Manchester Output Size.** 0: Manchester encode operations generate a half-word output, and decode operations generate a byte output. 1: Manchester encode operations generate a word output, and decode operations generate a half-word output.
3	RESET	**Module Reset.** Setting this bit to 1 flushes all data out of the module. No register settings are affected by the reset.
2	ERRIEN	**Error Interrupt Enable.** 0: Disable the error interrupt. 1: Enable the error interrupt.
1	ORDYIEN	**Output Ready Interrupt Enable.** 0: Disable the output ready interrupt. 1: Enable the output ready interrupt.
0	INRDYIEN	**Input Ready Interrupt Enable.** 0: Disable the input ready interrupt. 1: Enable the input ready interrupt.

Register 21.2. ENCDEC0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											DORI	DURI	DERRI	ORDYI	INRDYI
Type	R											RW	RW	RW	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register ALL Access Address

ENCDEC0_STATUS = 0x4004_F010

Table 21.8. ENCDEC0_STATUS Register Bit Descriptions

Bit	Name	Function
31:5	Reserved	Must write reset value.
4	DORI	**Data Overrun Interrupt Flag.** Hardware sets this bit to 1 when an input data overrun condition occurs and can cause an interrupt if enabled (ERRIEN = 1). This flag must be cleared by firmware.
3	DURI	**Data Underrun Interrupt Flag.** Hardware sets this bit to 1 when an output data underrun condition occurs and can cause an interrupt if enabled (ERRIEN = 1). This flag must be cleared by firmware.
2	DERRI	**Data Error Interrupt Flag.** Hardware sets this bit to 1 when an illegal byte or invalid data is written to the DATAIN register and can cause an interrupt if enabled (ERRIEN = 1). This flag must be cleared by firmware.
1	ORDYI	**Output Ready Interrupt Flag.** Hardware sets this bit to 1 when the output data in DATAOUT is ready and can cause an interrupt if enabled (ORDYIEN = 1). Reading the DATAOUT or DATAOUTC registers will automatically clear this flag. This flag is automatically cleared by hardware when operating in DMA mode (DMAEN = 1).
0	INRDYI	**Input Ready Interrupt Flag.** Hardware sets this bit to 1 when the input data FIFO is empty and can be written with data. This bit can also cause an interrupt if enabled (INRDYIEN = 1). Writing to the DATAIN register will automatically clear this flag. This flag is automatically cleared by hardware when operating in DMA mode (DMAEN = 1).

Register 21.3. ENCDEC0_DATAIN: Data Input

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATAIN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATAIN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ENCDEC0_DATAIN = 0x4004_F020

Table 21.9. ENCDEC0_DATAIN Register Bit Descriptions

Bit	Name	Function
31:0	DATAIN	**Data Input.** This field is the input data to the ENCDEC module. A write to this register will initiate the encode or decode operation specified by the OPMD and EDMD bits in the order specified by the IORDER field. Hardware will set the INRDYI flag when the operation completes and new data can be written.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

Register 21.4. ENCDEC0_DATAOUT: Data Output

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATAOUT[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATAOUT[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

ENCDEC0_DATAOUT = 0x4004_F030

Table 21.10. ENCDEC0_DATAOUT Register Bit Descriptions

Bit	Name	Function
31:0	DATAOUT	**Data Output.** This module writes the encoded or decoded output data to this field in the order specified by the OORDER field and sets the ORDYI bit. A read of this field automatically clears the ORDYI bit. All available data must be read in a single operation.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

Register 21.5. ENCDEC0_DATAOUTC: Data Output Complement

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATAOUTC[31:16]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATAOUTC[15:0]															
Type	R															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

ENCDEC0_DATAOUTC = 0x4004_F040

Table 21.11. ENCDEC0_DATAOUTC Register Bit Descriptions

Bit	Name	Function
31:0	DATAOUTC	**Data Output Complement.** A read of this field will return the 1's complement of the DATAOUT field. A read of this field automatically clears the ORDYI bit.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

21.9. ENCDEC0 Register Memory Map

Table 21.12. ENCDEC0 Memory Map

Register Name	ENCDEC0_CONTROL	ENCDEC0_STATUS	ENCDEC0_DATAIN
ALL Address	0x4004_F000	0x4004_F010	0x4004_F020
Access Methods	ALL | SET | CLR	ALL	ALL
Bit 31	Reserved	Reserved	DATAIN
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15	IORDER		
Bit 14			
Bit 13	OORDER		
Bit 12			
Bit 11	Reserved		
Bit 10	DBGMD		
Bit 9	DMAEN		
Bit 8	BEN		
Bit 7	Reserved		
Bit 6	OPMD		
Bit 5	EDMD		
Bit 4	MOSIZE	DORI	
Bit 3	RESET	DURI	
Bit 2	ERRIEN	DERRI	
Bit 1	ORDYIEN	ORDYI	
Bit 0	INRDYIEN	INRDYI	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 21.12. ENCDEC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ENCDEC0_DATAOUT	0x4004_F030	ALL	DATAOUT																															
ENCDEC0_DATAOUTC	0x4004_F040	ALL	DATAOUTC																															

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

22. Enhanced Programmable Counter Array (EPCA0)

This section describes the Enhanced Programmable Counter Array (EPCA) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the EPCA block, which is used by all device families covered in this document.

22.1. Enhanced Programmable Counter Array Features

The Enhanced PCA module is a multi-purpose counter array with features designed for motor control applications. The EPCA module includes:

- Three sets of channel pairs (six channels total) capable of generating complementary waveforms.
- Center- and edge-aligned waveform generation.
- Programmable dead times that ensure channel pairs are never both active at the same time.
- Programmable clock divisor and multiple options for clock source selection.
- Waveform update scheduling.
- Option to function while the core is inactive.
- Multiple synchronization triggers and outputs to synchronize with other blocks in the device, such as the SARADCs.
- Pulse-Width Modulation (PWM) waveform generation.
- High-speed square wave generation.
- Input capture mode.
- DMA capability for both input capture and waveform generation.

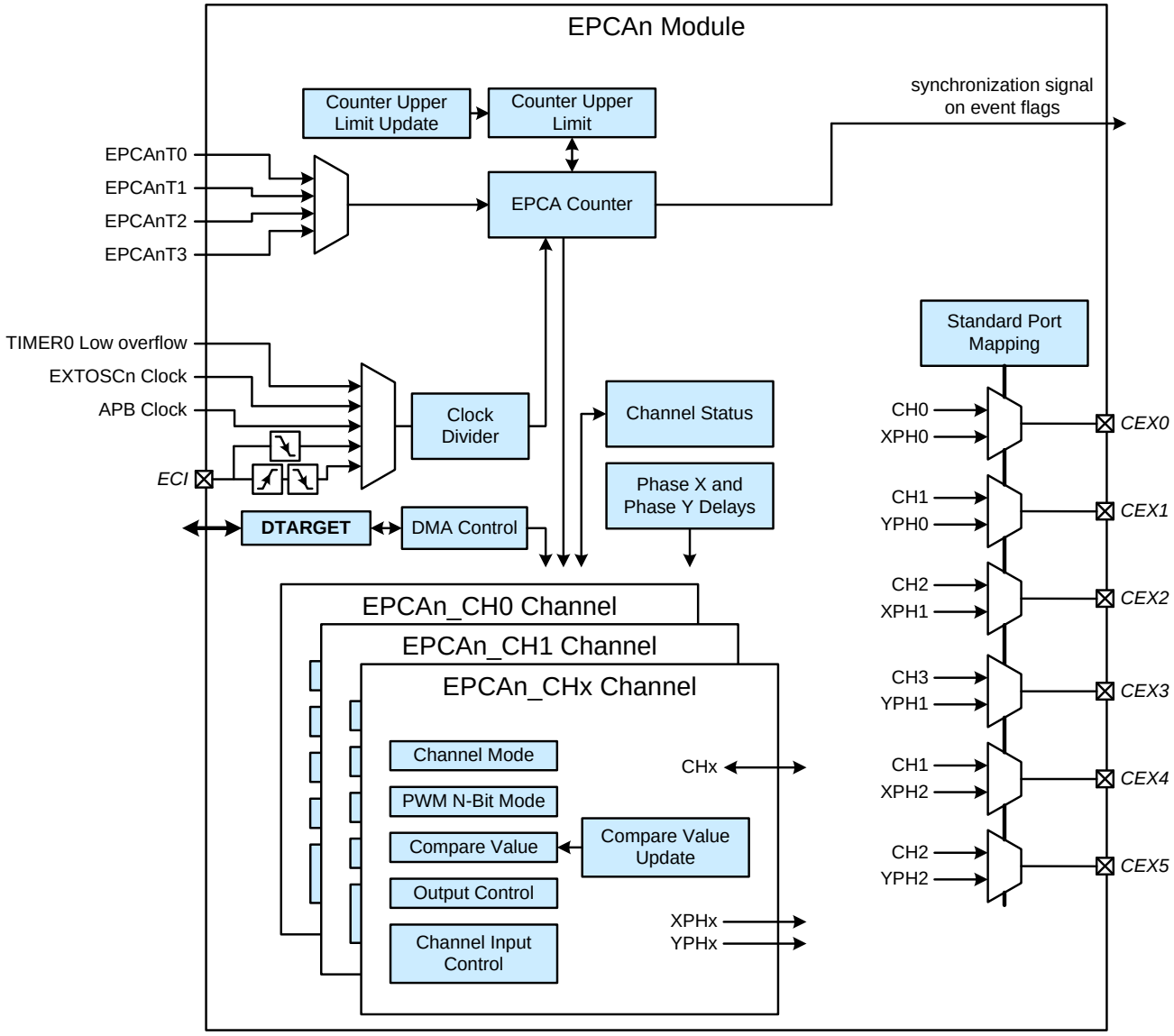


Figure 22.1. EPCA Block Diagram

22.2. Module Overview

The Enhanced Programmable Counter Array (EPCA) provides flexible timer/counter functionality. The EPCA consists of a dedicated 16-bit counter/timer and multiple (up to six) 16-bit capture/compare channels (EPCA0_CHx). All channels trigger or capture based on the shared 16-bit counter/timer, so the channels are inherently synchronized. The counter/timer is a 16-bit up counter with a programmable upper count limit. The counter starts at 0 and counts continuously from zero to the upper limit. Each channel includes a data register that can compare against the counter or capture the state of the counter based on a set of programmable conditions.

Each channel has its own associated output lines and may be configured to operate independently of the others in one of six modes: Edge-Aligned PWM, Center-Aligned PWM, High-Frequency / Square Wave, Timer / Capture, n-bit Edge-Aligned PWM, or Software Timer. The output lines may be single (one per channel, CHx) or differential (a complimentary pair per channel, XPHx and YPHx). These signals are then mapped to the STD_CEXx EPCA signals based on the STDOSEL fields.

The RUN bit starts or stops the EPCA counter, but there are also additional controls that can halt the counter. Additionally, the DBGMD bit in the CONTROL register controls whether the EPCA counter runs while the core is halted in debug mode.

The EPCA additionally has up to four external trigger signals that can start the counter. The STSEL field in the MODE register determines which of the trigger sources can start the counter. STESEL selects the active polarity of the external signal, and setting STEN to 1 enables the external signal to start the counter. Once the counter is running, the external trigger will not stop the counter.

Three different events can change a channel output or cause a capture. An overflow/limit event occurs when the EPCA counter (COUNTER) is equal to the upper limit (LIMIT). A capture/compare event occurs whenever a channel's capture/compare register (CCAPV) matches the current EPCA counter value. An intermediate overflow event can only occur in n-bit PWM mode and occurs when the channel's n-bit capture/compare range matches the EPCA counter.

To facilitate counter and channel updates while the counter continues to run, the counter upper limit register (LIMIT) and channel compare/capture register (CCAPV) have update registers (LIMITUPD and CCAPVUPD, respectively). Firmware can write new values to the LIMITUPD and CCAPVUPD registers without modifying the current EPCA cycle. The hardware will load these values into the LIMIT and CCAPV registers when the next counter overflow/limit event occurs, and the module UPDCF and channel CUPDCF flags indicate when an update operation completes. Firmware can set the NOUPD bit to 1 to prevent updates from occurring.

22.3. Clocking

The clock input for the counter/timer is selected by the CLKSEL field in the MODE register as shown in Figure 22.2.

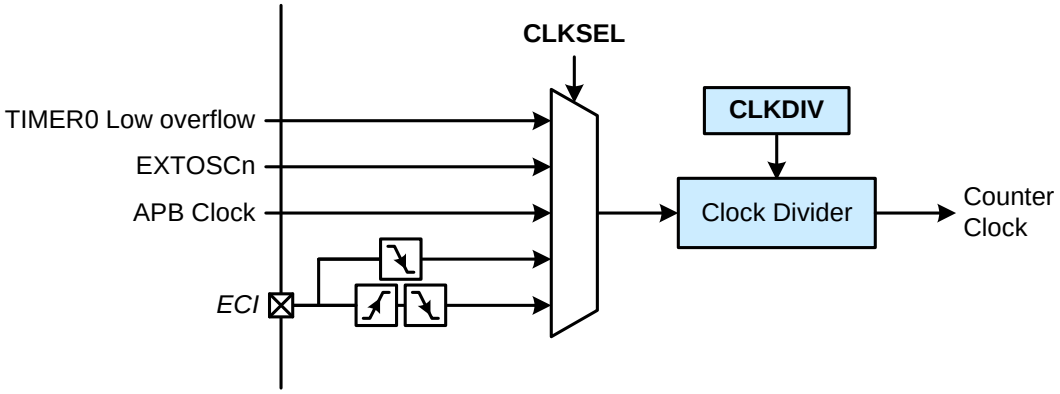


Figure 22.2. Clock Source Selection

The selected clock is passed through a divide-by-n clock divider where n can be between 1 and 1024. The CLKDIV field sets the clock divider for the selected clock before it drives the 16-bit counter/timer.

All non-APB clock sources are synchronized with the APB clock, so the maximum possible operating speed using these sources is one-half the APB frequency.

22.3.1. APB Clock

When the APB clock signal is selected as the counter clock source, the EPCA counter clock divider will use the APB clock source defined by the device clock control module. Selecting the APB clock provides the fastest clock source for the module.

22.3.2. External Clock (EXTOSCn)

When the external clock is selected as the counter clock source, the counter will run from the external clock source (EXTOSCn), regardless of the clock selection of the core. The external clock source is synchronized to the APB clock in this mode. In order to guarantee that the external clock transitions are recognized by the device, the external clock signal must be high or low for at least one APB clock period. This limits the maximum frequency of an external clock in this mode to one-half the APB clock.

22.3.3. TIMER0 Low Overflow

The EPCA module can select the TIMER0 module low timer overflows as its clock source. TIMER0 can operate in either 32-bit or 16-bit mode and still provide the clock for the EPCA clock divider. The maximum speed for the TIMER0 low overflows as an EPCA counter clock source is one-half the APB clock.

22.3.4. External Clock Input (ECI)

When the external clock input (ECI) is selected as the EPCA clock source, the clock divider decrements on falling edges or both rising and falling edges of the pin. The ECI pin is synchronized to the APB clock in this mode. The maximum clock rate for the ECI external clock input is the APB divided by 4.

22.3.5. Clock Divider

The clock divider provides a flexible time base for the EPCA counter. The divider starts at one-half the CLKDIV value and decrements to 0. Using this method, the divider counts the number of input clocks until the next counter clock edge (either rising or falling) rather than whole counter clock periods.

The current value of the divider can be read and written using the DIV field in the MODE register. This allows access during debugging to observe module events at the various EPCA clock edges rather than stepping through a large number of APB clocks. Firmware should always write to the CLKDIV field to modify the divider value.

The DIVST bit displays the current output phase of the clock divider.

22.4. Interrupts

The EPCA module has one interrupt vector and multiple interrupt sources within the module.

The OVFIEN bit enables the counter overflow/limit interrupt, which occurs when counter (COUNTER) equals the upper limit (LIMIT) and resets to 0. The OVFI interrupt flag indicates when an counter overflow/limit event occurs.

Each channel (CHx) has a compare/capture interrupt flag (CxCCI) and an intermediate overflow flag (CxIOVFI). These interrupts can be enabled in the channel registers (CCIEN for capture/compare, CIOVFIEN for intermediate overflow).

Firmware can check the source of the EPCA interrupt by checking the appropriate flags in the interrupt service routine.

22.5. Outputs

The EPCA module has up to six module outputs (CEXx) that can be routed to physical pins by configuring the device port configuration module.

Each EPCA capture/compare channel has two independent output modes: single and differential. In single output mode, the channel has one CHx output. In differential mode, the channel has two XPHx and YPHx outputs. The mapping of these channel outputs to the EPCA module outputs (STD_CEXx) is determined by the STDOSEL field.

The STDOSEL field determines the mapping for standard port banks as routed through the crossbar. The mappings are selected such that up to three potential channels can operate in differential mode at the same time.

Table 22.1 shows the output mapping for the pins according to the STDOSEL value.

Table 22.1. Output Mapping

EPCA Output	STDOSEL Value			
	0	1	2	3
STD_CEX0	CH0	CH0	CH0	XPH0
STD_CEX1	CH1	CH1	CH1	YPH0
STD_CEX2	CH2	CH2	XPH1	XPH1
STD_CEX3	CH3	CH3	YPH1	YPH1
STD_CEX4	CH4	XPH2	XPH2	XPH2
STD_CEX5	CH5	YPH2	YPH2	YPH2

Note: The COUTST (and optionally XPHST, YPHST, and ACTIVEPH) bit determines the starting state of the channel output. If the starting state is active, this means the channel could be sitting in an active state for some time while firmware finishes initializing the module and starts the counter. If the output is connected to a transistor where this behavior is undesireable, firmware can initialize the counter to a mid-range value and the outputs with an inactive value. This will ensure that any sensitive external circuits will not be damaged.

22.5.1. Single Output Mode

When a channel operates in single output mode, the COUTST bit in the EPCA channels determines the polarity of the output. This value should be set when the counter is not running to ensure predictable operation. Hardware sets the COUTST bit on the rising edges of the APB clock to reflect the current output state of the channel.

Figure 22.3 shows an EPCA channel single output timing diagram.

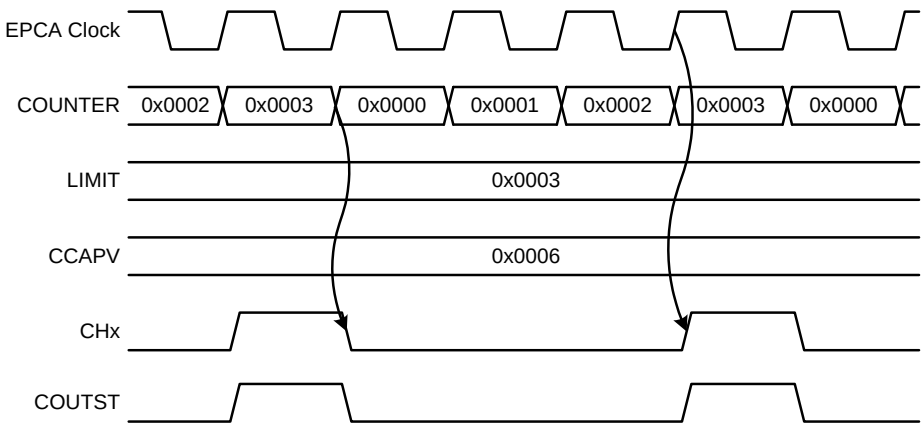


Figure 22.3. Example Channel Single Output Timing Diagram (Toggle Mode)

The CHx output can toggle, set, or clear on an overflow/limit, compare, or intermediate overflow event. In addition, the output can ignore these events and stay at its previous value. The COSEL field in the channel MODE register determines the CHx output behavior. This field allows firmware to modify the behavior of the CHx output without changing the configuration of the channel, interrupting the counter, or changing the port configuration.

22.5.2. Differential Output Mode

A channel additionally generates two synchronous outputs when operating in differential mode (DIFGEN = 1).

The YPHST and XPHST bits in the EPCA channels determine the polarity of the outputs. These values should be set when the counter is not running to ensure predictable operation. These bits can be read at any time to determine the current states of the outputs.

In addition, the differential outputs have programmable dead-time delays (AHB clocks) that prevent both channels from being active at the same time. The X delay time (DTIMEX) starts when the XPHx output switches to its inactive state. The YPHx output then switches to its active state when the time expires, and the Y delay time (DTIMEY) behaves similarly. These delay times are global for all channels.

The channel ACTIVEPH bit indicates which output is currently active. The active channel can be asserted or deasserted pending the dead time delay timeout.

Figure 22.4 shows an EPCA channel differential output timing diagram.

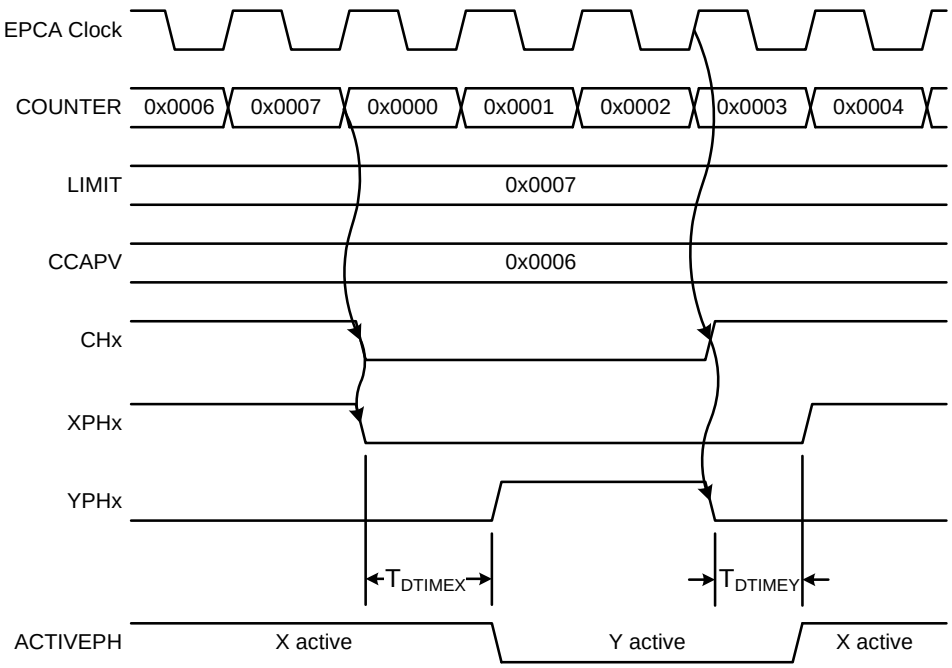


Figure 22.4. Example Channel Differential Output Timing Diagram

The output behavior of the XPHx and YPHx outputs depends on the behavior of the CHx output. These phase outputs will toggle as long as CHx is toggling. If the COSEL field is set such that the CHx output is no longer toggling (set, clear, or ignore), the XPHx and YPHx outputs will not change state. As long as the hardware control of the CHx output remains static, firmware can manually stimulate the XPHx and YPHx outputs by writing COUTST to different states and directly controlling the CHx output.

22.5.3. Synchronization Signal

In addition to the CHx, XPHx, and YPHx outputs, the EPCA module can generate a synchronization signal for use by other modules on the device (SARADC0, TIMER0, TIMER1, or TIMER2). This signal can pulse when a counter overflow/limit (OVFSEN = 1), channel intermediate overflow (CIOVFSEN = 1), or channel capture/compare (CCSEN = 1) event occurs.

22.6. Triggers

The EPCA supports four trigger sources. The selections for the trigger are made in the STSEL, STESEL and STEN fields of the CONTROL register, The trigger sources for SiM3L1xx devices are defined in Table 22.2.

Table 22.2. EPCA0 Triggers

EPCA0 Trigger	EPCA0 Trigger Description	Internal Signal
EPCA0T0	Internal Trigger Source	Comparator 0 output
EPCA0T1	Internal Trigger Source	Comparator 1 output
EPCA0T2	Internal Trigger Source	Timer 0 High Overflow output
EPCA0T3	Internal Trigger Source	Timer 1 High Overflow output

To use a trigger:

1. Set up the desired trigger source module or channel.
2. Select the trigger source using the STSEL field.
3. Set the polarity of the trigger using the STESEL bit.
4. Enable the trigger (STEN = 1).
5. Set up the desired EPCA counter and channel settings.
6. Start the EPCA counter by setting RUN to 1.

The EPCA counter will start running when the selected trigger source meets the criteria set by the STESEL polarity. The counter will then continue to run regardless of the state of the trigger source.

22.7. Operational Modes

The EPCA module has six operational modes that each channel can independently select: Edge-Aligned PWM, Center-Aligned PWM, High-Frequency/Square Wave, Timer/Capture, n-bit Edge-Aligned PWM, and Software Timer modes. The CMD bits select the channel operational mode.

The EPCA counter is 16 bits and counts in full EPCA clock cycles. The channel CCAPV registers are 18 bits and represent half EPCA clock cycles. To match a counter value, the CCAPV field should be written with double the counter value. The CCAPV LSB provides timing resolution to one half-clock counter cycle. The additional 18th bit of the CCAPV register allows the channel outputs to create 0-100% duty cycles (0x00000 is 0% and 0x20000 is 100% duty cycle).

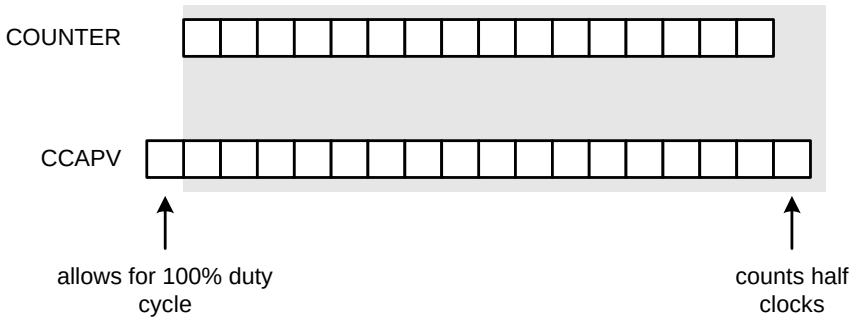


Figure 22.5. EPCA Counter and Channel Compare/Capture Registers

This section discusses the channel behavior in each of these modes in detail.

22.7.1. Edge-Aligned Pulse Width Modulation (PWM) Mode

In edge-aligned PWM mode (CMD = 0), the 18-bit capture/compare register (CCAPV) defines the number of EPCA half clocks for the inactive time of the PWM signal. A capture/compare event occurs when the counter matches the register contents, and the output will change from the initial state set by COUTST depending on the selected COSEL value (typically 00b for toggle). An overflow/limit event occurs when the counter reaches the LIMIT value and resets to 0, and CHx will again change depending on the COSEL value. To output a varying duty cycle, firmware can write to the CCAPVUPD register, which hardware will automatically load into the channel's CCAPV register on the next counter overflow/limit if the module register update inhibit (NOUPD) is cleared to 0.

Assuming that the inactive and initial state of the output is low and COSEL is set to toggle, the CHx output duty cycle in edge-aligned PWM mode is shown in Equation 22.1. No output pulse is generated if CCAPV is 0, and 100% duty cycle results when CCAPV is set to 0x20000. The resulting XPHx and YPHx timing also depends on the dead-time delay values.

Figure 22.6 shows an example edge-aligned PWM timing diagram.

$$\text{Duty Cycle} = \frac{((\text{LIMIT} + 1) \times 2) - \text{CCAPV}}{(\text{LIMIT} + 1) \times 2}$$

Equation 22.1. Edge-Aligned PWM Duty Cycle

Because CCAPV is given in half clocks, an odd CCAPV value results in the CHx output changing state at the mid-cycle edge of the EPCA clock.

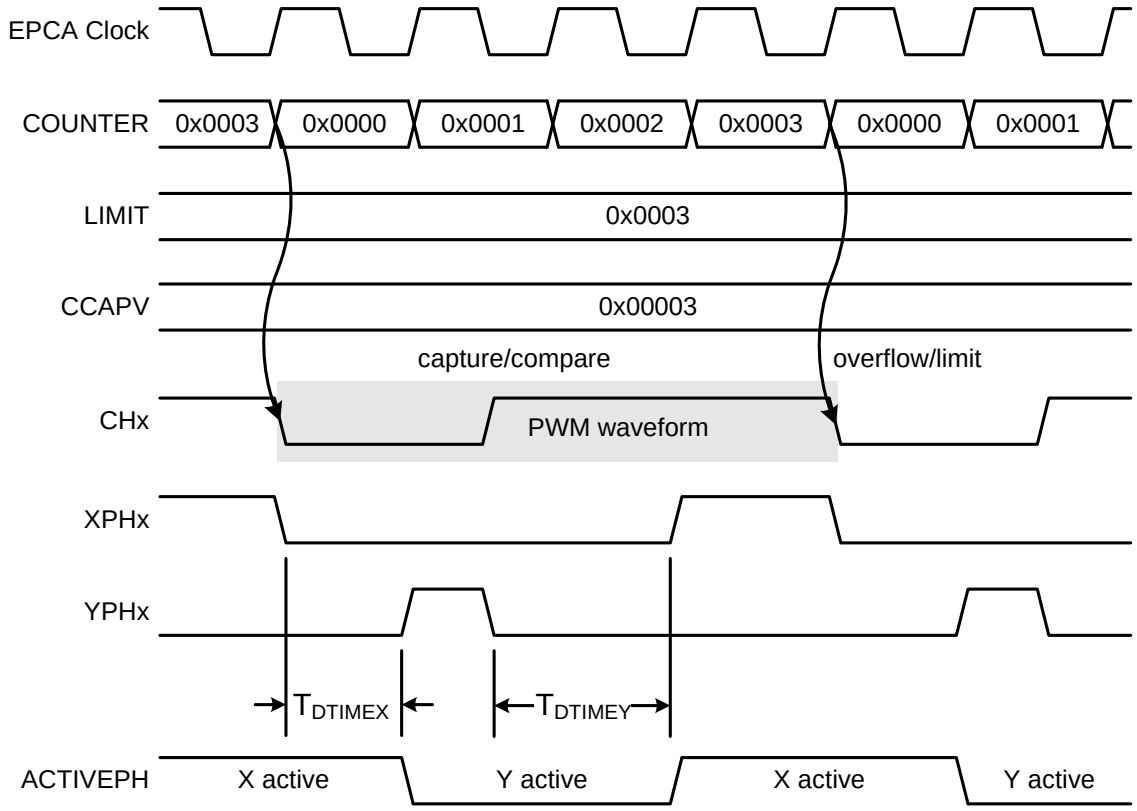


Figure 22.6. Example Edge-Aligned PWM Timing Diagram

22.7.2. Center-Aligned Pulse Width Modulation (PWM) Mode

In center-aligned PWM mode (CMD = 1), a channel generates a PWM waveform symmetric about the counter's overflow/limit event defined by the LIMIT field. Multiple channels in an array configured in this mode will each have PWM waveforms which are all symmetric about the same center-pulse points (counter equal to zero).

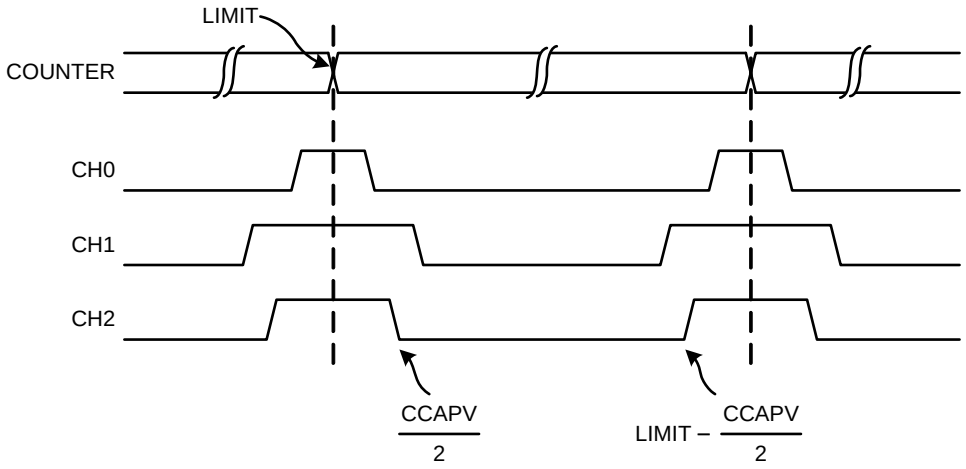


Figure 22.7. Multiple Center-Aligned PWM Channels

The channel 18-bit capture/compare register (CCAPV) is used to generate two capture/compare events in one counter cycle (0 to LIMIT). The first event occurs when the counter is equal to CCAPV divided by 2. The second event occurs when the counter is equal to LIMIT minus CCAPV divided by 2. In the event of an odd value in CCAPV, the hardware adds the extra half cycle to the capture/compare event following the counter overflow/limit event.

Firmware can write to the channel's CCAPVUPD register to update the waveform. Hardware will update the CCAPV field with the CCAPVUPD value when the counter overflows from the upper limit to zero as long as the update inhibit bit (NOUPD) is cleared to 0.

The COUTST (and optionally XPHST, YPHST, and ACTIVEPH) bit determines the starting state of the channel output and should be set before starting the counter. If the starting state is active, this means the channel could be sitting in an active state for some time while firmware finishes initializing the module and starts the counter. If the output is connected to a transistor where this behavior is undesireable, firmware can initialize the counter to a mid-range value and the outputs with an inactive value. This will ensure that any sensitive external circuits will not be damaged.

Assuming that the active and initial state of the output is high and COSEL is set to toggle, the CHx output duty cycle in center-aligned PWM mode is shown in Equation 22.2. No output pulse is generated if CCAPV is 0, and 100% duty cycle results when CCAPV (in half clocks) is set to a number of full clocks equal to or greater than LIMIT. The resulting XPHx and YPHx timing also depends on the dead-time delay values.

Figure 22.8 shows an example center-aligned PWM timing diagram.

$$\text{Duty Cycle} = \frac{((\text{LIMIT} + 1) \times 2) - \text{CCAPV}}{(\text{LIMIT} + 1) \times 2}$$

Equation 22.2. Center-Aligned PWM Duty Cycle

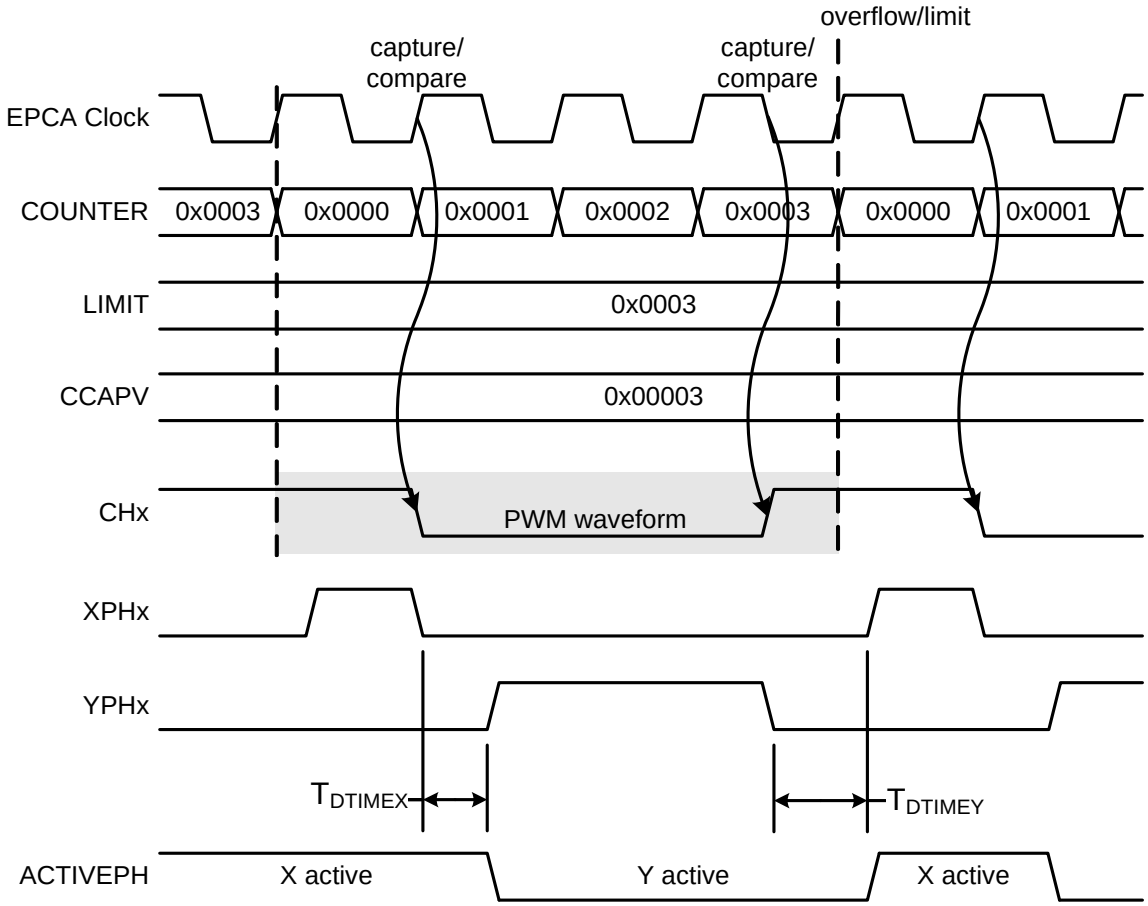


Figure 22.8. Example Center-Aligned PWM Timing Diagram

22.7.3. High-Frequency / Square Wave Mode

High-frequency square-wave mode (CMD = 2) produces a 50% duty cycle waveform of programmable period on the channel's CHx output. This mode provides a flexible way of creating square waves with a fast period. Each half-period of the generated output clock can be as short as one counter clock period (two CCAPV half clocks) and as long as 128 counter clock periods, programmable in half clock steps.

Bits [15:8] of the channel capture/compare register (CCAPV) contain the waveform half period in the number of half EPCA clocks. The lower byte (bits [7:0]) is the calculated value compared to the counter. When a match occurs between the lower byte and the counter, the hardware triggers a capture/compare event and automatically adds the match value of the counter to the CCAPV[15:8] value to create a new compare value for the lower byte. An overflow/limit event occurs when the counter reaches the upper limit defined by the LIMIT field. Firmware should program the upper limit register to reset the counter at a (multiple of 128) - 1 to avoid undesired waveform edges. Figure 22.9 shows how the hardware creates the waveform.

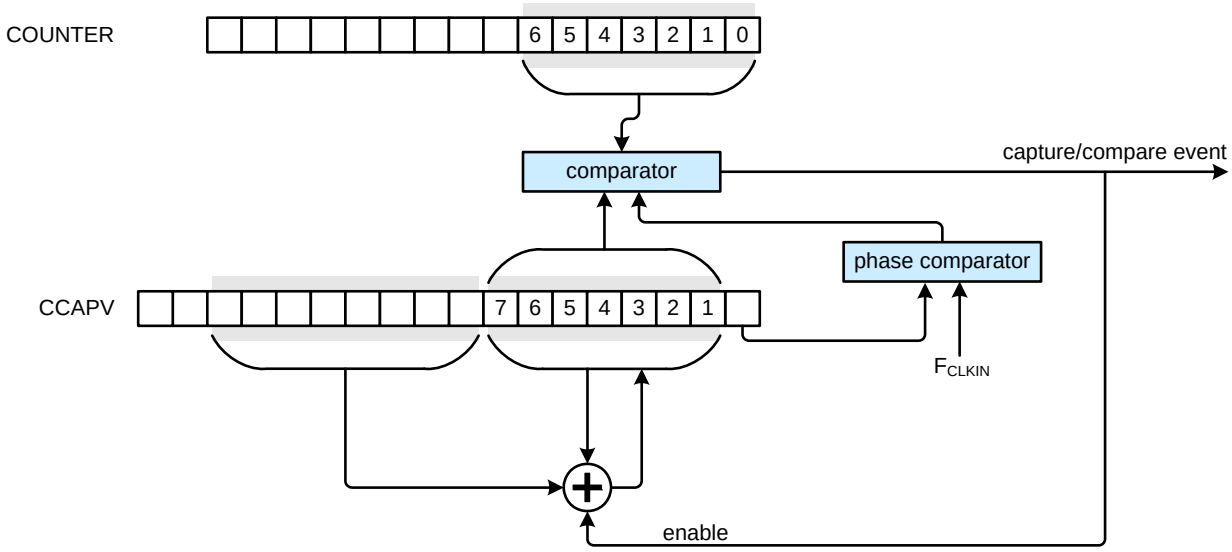


Figure 22.9. High-Frequency Square Wave Waveform Generation

Firmware can update the frequency of the output by writing to the CCAPVUPD register. Hardware will automatically load bits [15:8] of this register to the CCAPV register at the next counter overflow/limit event, if possible (NOUPD = 0). The rest of the CCAPVUPD register (bits [17:16] and [7:0]) are ignored.

Assuming that the COSEL field for the channel is set to toggle, the resulting CHx output frequency in high-frequency square-wave mode is shown in Equation 22.3. A CCAPV[15:8] value of 1 is not valid, and a CCAPV[15:8] value of 0 results in an output waveform half period of 128 counter clocks. The resulting XPHx and YPHx timing also depends on the dead-time delay values.

Figure 22.10 shows an example high-frequency square wave mode timing diagram.

$$F_{CHx} = \frac{F_{EPCA}}{CCAPV[15:8]}$$

Equation 22.3. High-Frequency Square Wave Output Frequency

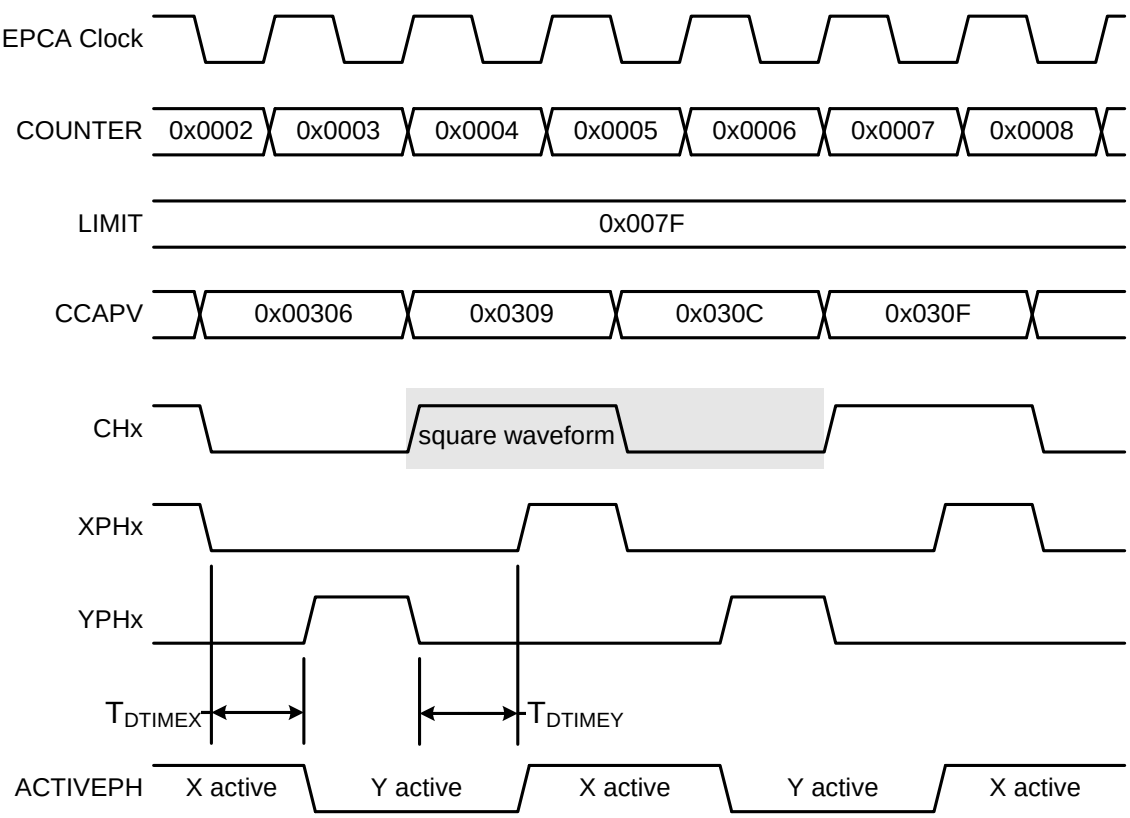


Figure 22.10. Example High-Frequency Square Wave Mode Timing Diagram

22.7.4. Timer / Capture Mode

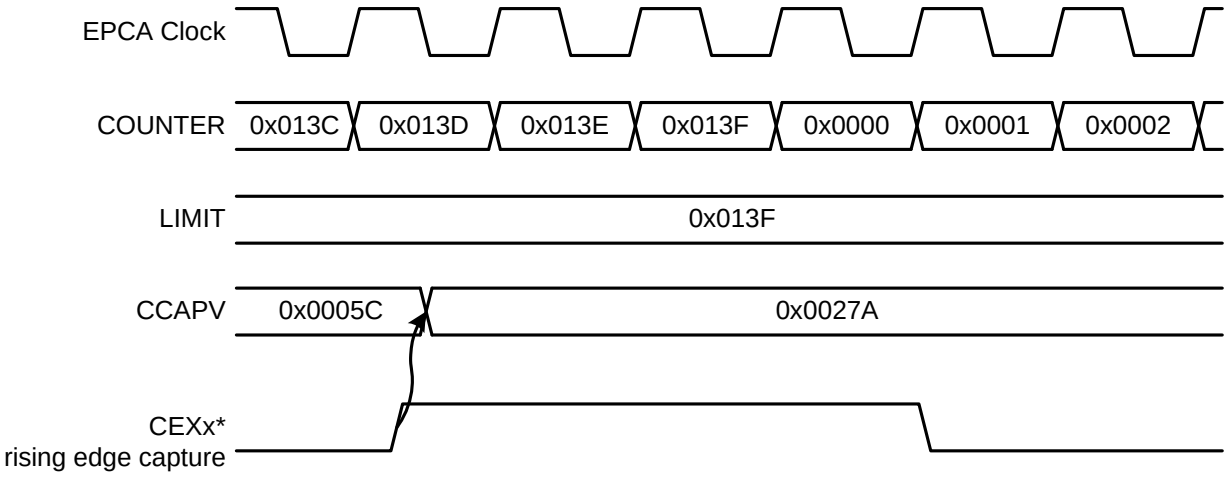
In timer/capture mode (CMD = 3), the channel uses the CEXn line as an input signal that triggers a capture/compare event and stores the counter state in the channel capture/compare register (CCAPV) in half clocks. Firmware can configure the event to trigger on rising, falling, or both edges using the channel CPCAPEN and CNCAPEN bits. Table 22.3 shows the capture edge configuration options.

Table 22.3. Capture Edge Configuration Options

CPCAPEN	CNCAPEN	Selected Edge
0	0	Capture disabled.
0	1	Capture on the falling edge.
1	0	Capture on the rising edge.
1	1	Capture on both edges.

The input CEXn signal must remain high or low for at least two APB clocks to be recognized by the hardware. The capture occurs at the next EPCA clock edge.

Figure 22.11 shows an example timer/capture mode timing diagram.



*Note: CEXx must be high or low for at least 2 APB clock cycles.

Figure 22.11. Example Timer/Capture Mode Timing Diagram

22.7.5. N-bit Edge-Aligned Pulse Width Modulation (PWM) Mode

The n-bit edge-aligned PWM modes allow each channel to be independently configured to generate edge-aligned PWM waveforms with a faster duty cycle than the counter. In n-bit edge-aligned PWM mode (CMD = 4), the least-significant n bits (set by PWMMD) of the counter define the number of full clocks of the PWM waveform. The channel's capture/compare register (CCAPV) defines the number of EPCA half clocks for the inactive time of the PWM signal, and the higher order unused bits of CCAPV are ignored. A capture/compare event occurs when the n-bit counter is equal to the number of half clocks defined by the CCAPV field. An intermediate overflow event occurs when the counter overflows the n-bit range. A counter overflow/limit event occurs when the counter reaches the upper limit (LIMIT) within the full 16-bit range. If one of the channels operates in n-bit mode, firmware should set the counter's upper limit as an even multiple of the n-bit boundary to ensure the channel's output does not have any irregular edges from the overflow/limit events.

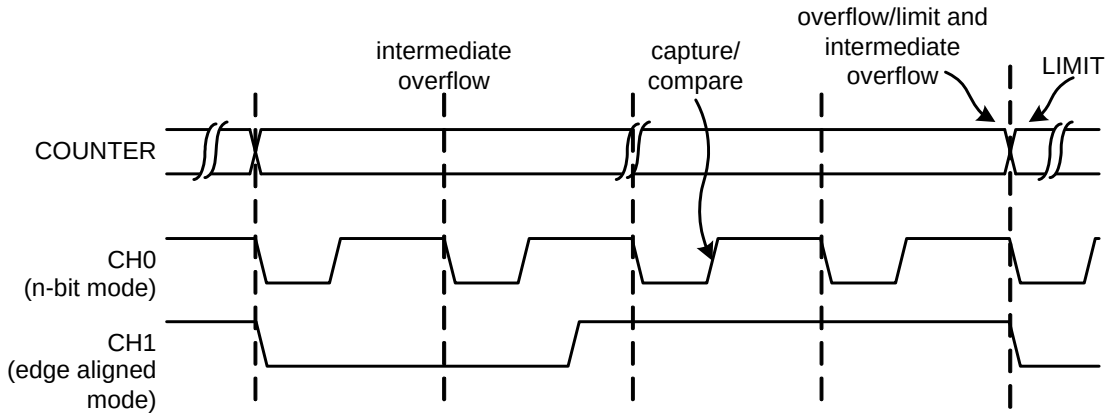


Figure 22.12. Multiple Edge-Aligned PWM Channels (N-bit and Edge-Aligned)

Table 22.4 provides a list of the intermediate overflow, recommended upper limit values, and 100% duty cycle values for each n-bit setting.

Table 22.4. N-bit Intermediate Overflow Values

PWMMD Value	N-bit PWM Mode	Counter Intermediate Overflow Value (full clocks)	Recommended Counter Upper Limit (full clocks)	CCAPV 100% Duty Cycle Value (half clocks)
0	0-bit	0	any value	2
1	1-bit	1	(multiple of 2) - 1	4
2	2-bit	3	(multiple of 4) - 1	8
3	3-bit	7	(multiple of 8) - 1	16
4	4-bit	15	(multiple of 16) - 1	32
5	5-bit	31	(multiple of 32) - 1	64
6	6-bit	63	(multiple of 64) - 1	128
7	7-bit	127	(multiple of 128) - 1	256
8	8-bit	255	(multiple of 256) - 1	512
9	9-bit	511	(multiple of 512) - 1	1024
10	10-bit	1023	(multiple of 1024) - 1	2048
11	11-bit	2047	(multiple of 2048) - 1	4096
12	12-bit	4095	(multiple of 4096) - 1	8192
13	13-bit	8191	(multiple of 8192) - 1	16334
14	14-bit	16333	(multiple of 16334) - 1	32768
15	15-bit	32767	(multiple of 32768) - 1	65536

Firmware can write to the CCAPVUPD register to update the duty cycle, which hardware will automatically load into the channel's CCAPV register on the next counter overflow/limit event if the module register update inhibit (NOUPD) is cleared to 0.

Assuming that the inactive and initial state of the output is low, COSEL is set to toggle, and the upper limit is set to an even multiple, the CHx output duty cycle in n-bit edge-aligned PWM mode is shown in Equation 22.4. No output pulse is generated if CCAPV is 0. The resulting XPHx and YPHx timing also depends on the dead-time delay values.

Figure 22.13 shows an example n-bit edge-aligned PWM timing diagram.

$$\text{Duty Cycle} = \frac{(2^n \times 2) - \text{CCAPV}}{2^n \times 2}$$

Equation 22.4. N-bit Edge-Aligned PWM Duty Cycle

Because CCAPV is given in half clocks, an odd CCAPV value results in the CHx output changing state at the mid-cycle edge of the EPCA clock.

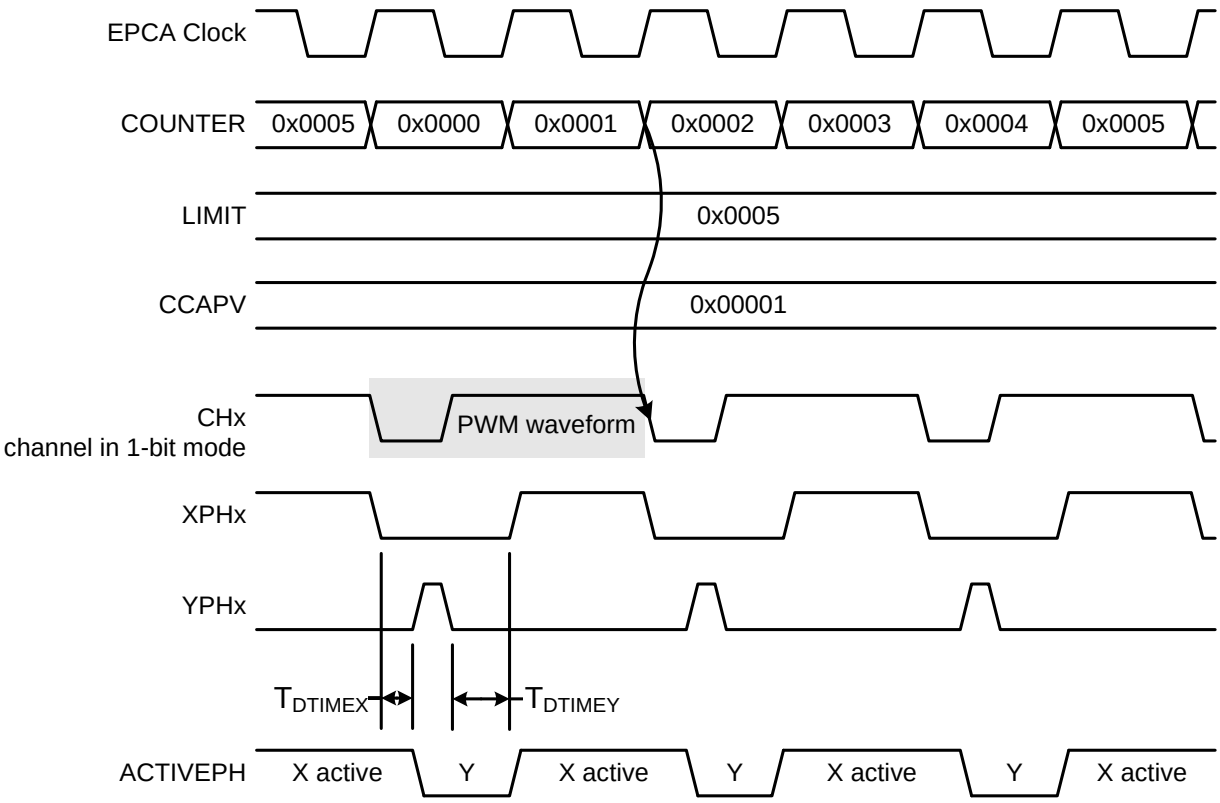


Figure 22.13. Example N-bit Edge-Aligned PWM Timing Diagram

22.7.6. Software Timer Mode

Software timer mode is a free-running mode with the CHx output unaffected by the counter or channel state. Setting the COSEL field to 3 (disable output events) enters software timer mode, regardless of the CMD field setting. The counter overflow/limit, intermediate overflow, and capture/compare events can still be used to generate interrupts, even though the channel output is disabled.

Figure 22.14 shows an example software timer mode timing diagram.

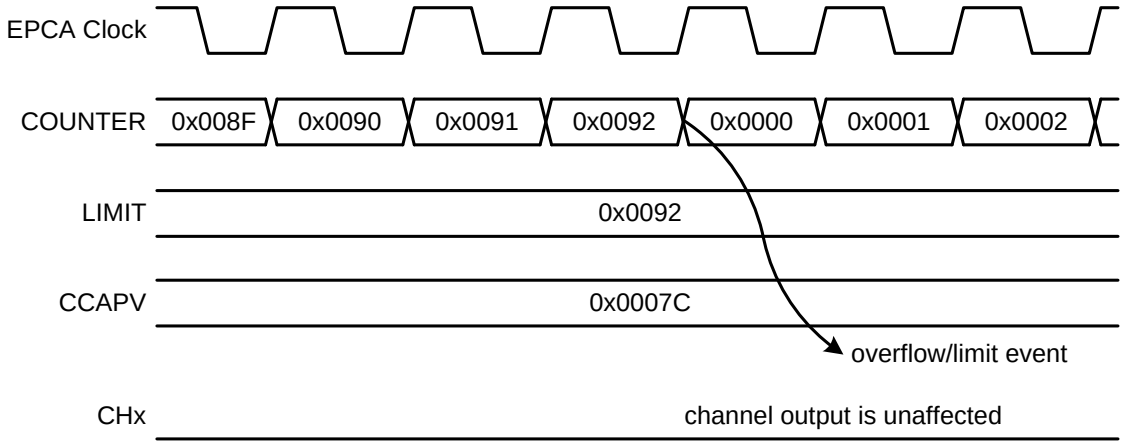


Figure 22.14. Example Software Timer Mode Timing Diagram

22.8. DMA Configuration and Usage

The EPCA supports two DMA channels: control and capture. The control requests move data from memory into the counter and update channel registers to autonomously set up new waveform generation. Capture requests move the counter capture data from the channel CCAPV register to memory.

The hardware can generate a DMA transfer request with a counter overflow/limit (OVFDEN = 1), channel intermediate overflow (CIOVFDEN = 1), or channel capture event (CCDEN = 1). When these events are enabled as a source for a DMA transfer, hardware automatically clears the flags associated with the request after the transfer completes. A DMA transfer to service a control request will not clear a flag for a pending capture service request.

The EPCA Module DMA configuration is shown in Figure 22.15.

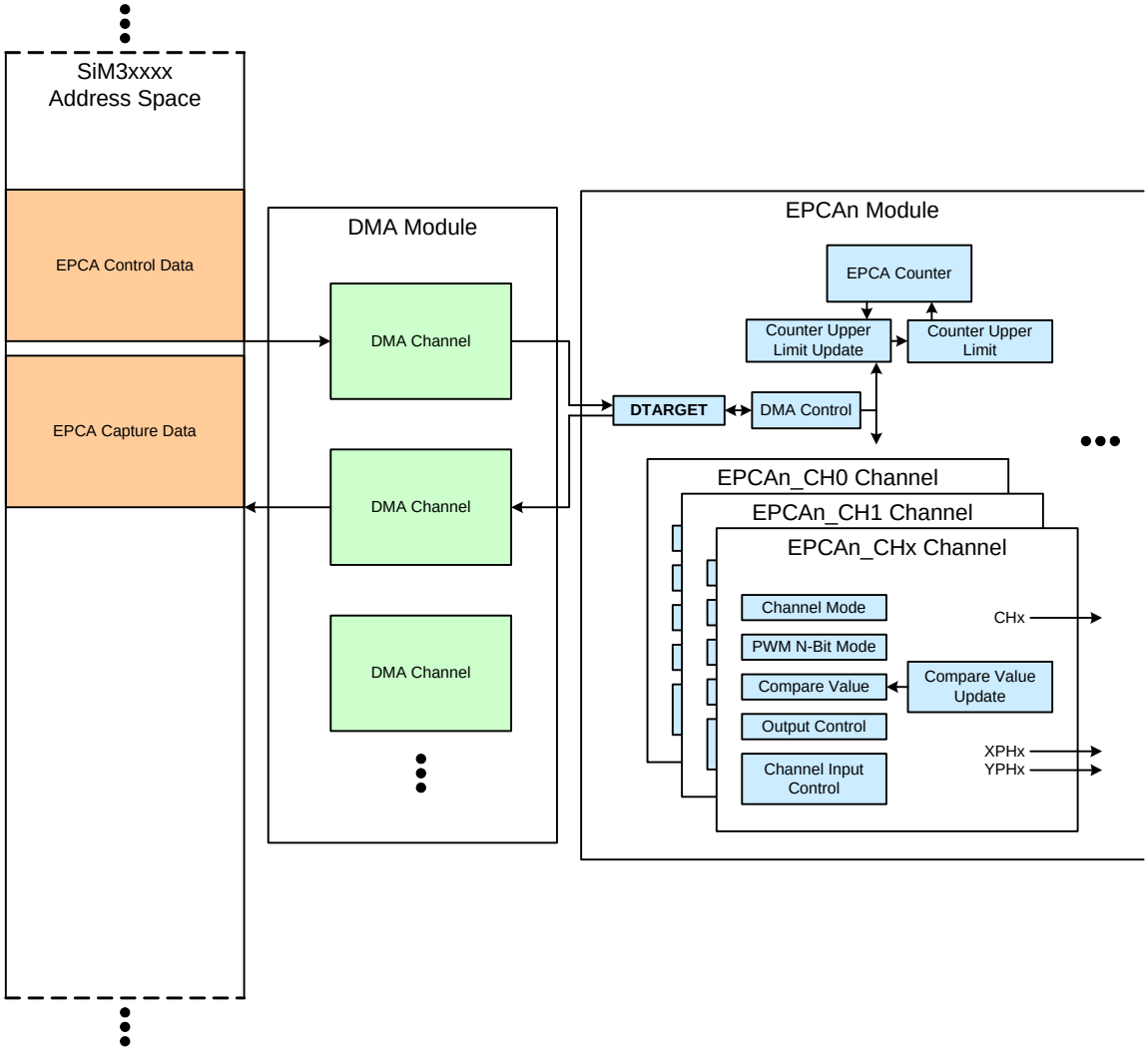


Figure 22.15. EPCA Module DMA Configuration

22.8.1. Control DMA Transfers

In a control transfer, the DMA should be configured to transfer one word at a time to the DTARGET location with the destination in non-incrementing mode. The module MODE register contains three pointers for use with a control transfer: DSTART, DEND, and DPTR. These pointers control the transfer of data from the DTARGET register to the module and channel update registers (LIMITUPD and CCAPVUPD). The DSTART pointer indicates the starting register in a new DMA write transfer, DEND indicates the ending register in the transfer, and DPTR is a circular pointer to the next register a DMA write to DTARGET will update.

Table 22.5. DMA Control Transfer Pointer Slots

DSTART, DEND, or DPTR Value	Register
0	LIMITUPD
1	Channel 0 CCAPVUPD
2	Channel 1 CCAPVUPD
3	Channel 2 CCAPVUPD
4	Channel 3 CCAPVUPD
5	Channel 4 CCAPVUPD
6	Channel 5 CCAPVUPD
7	empty

Slot 7 is not normally written unless DEND is set to 7, DSTART is set to 0, and the DMA uses 8-word transfers. In this case, the 8th data write is discarded.

A transfer starting at slot 5 (Channel 4 CCAPVUPD) can wrap around to end at an earlier slot. Any wraps ignore slot 7, so the sequence for a 4 DMA word transfers would be: slot 5, slot 6, slot 0, slot 1.

Firmware should not modify these fields while a DMA transfer is in progress, indicated by the DBUSYF flag.

22.8.2. Software DMA Transfers

Software DMA requests can be directly generated by two different software mechanisms:

1. Configure the EPCA for a DMA control request without enabling the DMA channel in the controller. Software can then write to the update registers by writing directly the DTARGET field.
2. Completely configure the DMA transfer (including the controller channel) and trigger DMA transfers by writing to the associated event interrupt flag.

22.9. EPCA0 Registers

This section contains the detailed register descriptions for EPCA0 registers.

Register 22.1. EPCA0_MODE: Module Operating Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved			STDOSEL		Reserved	DBUSYF	DSTART			DPTR			DEND		
Type	R			RW		R	RW	RW			RW			RW		
Reset	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			CLKSEL			CLKDIV									
Type	RW		R	RW			RW									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
EPCA0_MODE = 0x4000_E180

Table 22.6. EPCA0_MODE Register Bit Descriptions

Bit	Name	Function
31:29	Reserved	Must write reset value.
28:27	STDOSEL	**Standard Port Bank Output Select.** 00: Select the non-differential channel outputs (Channels 0-5) for the standard PB pins. 01: Select the differential output from Channel 2 and non-differential outputs from Channels 0, 1, 2, and 3 for the standard PB pins. 10: Select the differential outputs from Channels 1 and 2 and non-differential outputs from Channels 0 and 1 for the standard PB pins. 11: Select three differential outputs from Channels 0, 1, and 2 for the standard PB pins.
26	Reserved	Must write reset value.
25	DBUSYF	**DMA Busy Flag.** 0: The DMA channel is not servicing an EPCA control transfer. 1: The DMA channel is busy servicing an EPCA control transfer.

Table 22.6. EPCA0_MODE Register Bit Descriptions

Bit	Name	Function
24:22	DSTART	**DMA Target Start Index.** This field is the first register to be accessed in a new DMA write transfer set to DTARGET. This field should be written by software before the start of the DMA transfer. 000: Set the first register in a DMA write transfer to LIMITUPD. 001: Set the first register in a DMA write transfer to Channel 0 CCAPVUPD. 010: Set the first register in a DMA write transfer to Channel 1 CCAPVUPD. 011: Set the first register in a DMA write transfer to Channel 2 CCAPVUPD. 100: Set the first register in a DMA write transfer to Channel 3 CCAPVUPD. 101: Set the first register in a DMA write transfer to Channel 4 CCAPVUPD. 110: Set the first register in a DMA write transfer to Channel 5 CCAPVUPD. 111: Empty slot.
21:19	DPTR	**DMA Write Transfer Pointer.** This field is the current target of the DMA. The next word written to DTARGET will be transferred to the register selected by DPTR. This field is set by hardware and should not be modified by software. 000: The DMA channel will write to LIMITUPD next. 001: The DMA channel will write to Channel 0 CCAPVUPD next. 010: The DMA channel will write to Channel 1 CCAPVUPD next. 011: The DMA channel will write to Channel 2 CCAPVUPD next. 100: The DMA channel will write to Channel 3 CCAPVUPD next. 101: The DMA channel will write to Channel 4 CCAPVUPD next. 110: The DMA channel will write to Channel 5 CCAPVUPD next. 111: Empty slot.
18:16	DEND	**DMA Write End Index.** This field is the last register to be accessed in a DMA write transfer set. This field should be written by software before the start of the DMA transfer. 000: Set the last register in a DMA write transfer to LIMITUPD. 001: Set the last register in a DMA write transfer to Channel 0 CCAPVUPD. 010: Set the last register in a DMA write transfer to Channel 1 CCAPVUPD. 011: Set the last register in a DMA write transfer to Channel 2 CCAPVUPD. 100: Set the last register in a DMA write transfer to Channel 3 CCAPVUPD. 101: Set the last register in a DMA write transfer to Channel 4 CCAPVUPD. 110: Set the last register in a DMA write transfer to Channel 5 CCAPVUPD. 111: Empty slot.
15:13	Reserved	Must write reset value.
12:10	CLKSEL	**Input Clock (F_{CLKIN}) Select.** 000: Set the APB as the input clock (F_{CLKIN}). 001: Set Timer 0 low overflows divided by 2 as the input clock (F_{CLKIN}). 010: Set high-to-low transitions on ECI divided by 2 as the input clock (F_{CLKIN}). 011: Set the external oscillator module output (EXTOSCn) divided by 2 as the input clock (F_{CLKIN}). 100: Set ECI transitions divided by 2 as the input clock (F_{CLKIN}). 101-111: Reserved.

Table 22.6. EPCA0_MODE Register Bit Descriptions

Bit	Name	Function
9:0	CLKDIV	**Input Clock Divider.** The EPCA module clock is given by the equation: $F_{EPCA} = \frac{F_{CLKIN}}{CLKDIV + 1}$ Where the input clock (CLKIN) is determined by the CLKSEL bits.

Register 22.2. EPCA0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DIV										DIVST	Reserved				
Type	RW										RW	R				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	STEN	STESEL	STSEL		Reserved				DBGMD	Reserved	NOUPD	Reserved	OVFSEN	OVFDEN	OVFIEN
Type	R	RW	RW	RW		R	RW	R		RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EPCA0_CONTROL = 0x4000_E190

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 22.7. EPCA0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:22	DIV	**Current Clock Divider Count.** This field is the current value of the internal EPCA clock divider. The clock divider is a counter that starts at CLKDIV / 2 and counts down to zero.
21	DIVST	**Clock Divider Output State.** 0: The clock divider is currently in the first half-cycle. 1: The clock divider is currently in the second half-cycle.
20:15	Reserved	Must write reset value.
14	STEN	**Synchronous Input Trigger Enable.** 0: Disable the input trigger (EPCAnTx). The EPCA counter/timer will continue to run if the RUN bit is set regardless of the value on the input trigger. 1: Enable the input trigger (EPCAnTx). If RUN is set to 1, the EPCA counter/timer will start running when the selected input trigger (STSEL) meets the criteria set by STESEL. It will not stop running if the criteria is no longer met.
13	STESEL	**Synchronous Input Trigger Edge Select.** 0: A high-to-low transition (falling edge) on EPCAnTx will start the counter/timer. 1: A low-to-high transition (rising edge) on EPCAnTx will start the counter/timer.

Notes:

1. Because hardware updates the DIV and DIVST fields, the SET and CLR addresses are the only safe way to access the other fields in this register while the EPCA is running.

Table 22.7. EPCA0_CONTROL Register Bit Descriptions

Bit	Name	Function
12:11	STSEL	**Synchronous Input Trigger Select.** 00: Select input trigger 0, Comparator0 output. 01: Select input trigger 1, Comparator1 output. 10: Select input trigger 2, Timer 0 high overflow. 11: Select input trigger 3, Timer 1 high overflow.
10:7	Reserved	Must write reset value.
6	DBGMD	**EPCA Debug Mode.** 0: A debug breakpoint will stop the EPCA counter/timer. 1: The EPCA will continue to operate while the core is halted in debug mode.
5	Reserved	Must write reset value.
4	NOUPD	**Internal Register Update Inhibit.** 0: The EPCA registers will automatically load any new update values after an overflow/limit event occurs. 1: The EPCA registers will not load any new update values after an overflow/limit event occurs.
3	Reserved	Must write reset value.
2	OVFSEN	**EPCA Counter Overflow/Limit Synchronization Signal Enable.** The sychronization signal generated by the EPCA module can be used as an input by other modules to synchronize with a particular EPCA event or state. 0: Do not send a synchronization signal when a EPCA counter overflow/limit event occurs. 1: Send a synchronization signal when a EPCA counter overflow/limit event occurs.
1	OVFDEN	**EPCA Counter Overflow/Limit DMA Request Enable.** 0: Do not request DMA data when a EPCA counter overflow/limit event occurs. 1: Request DMA data when a EPCA counter overflow/limit event occurs.
0	OVFIEN	**EPCA Counter Overflow/Limit Interrupt Enable.** 0: Disable the EPCA counter overflow/limit event interrupt. 1: Enable the EPCA counter overflow/limit event interrupt.

Notes:

1. Because hardware updates the DIV and DIVST fields, the SET and CLR addresses are the only safe way to access the other fields in this register while the EPCA is running.

Register 22.3. EPCA0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	C5IOVFI	C4IOVFI	C3IOVFI	C2IOVFI	C1IOVFI	C0IOVFI	Reserved	UPDCF	OVFI	RUN	C5CCI	C4CCI	C3CCI	C2CCI	C1CCI	C0CCI
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EPCA0_STATUS = 0x4000_E1A0

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 22.8. EPCA0_STATUS Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15	C5IOVFI	**Channel 5 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 5 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 5 did not count past the channel n-bit mode limit. 1: Channel 5 counted past the channel n-bit mode limit.
14	C4IOVFI	**Channel 4 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 4 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 4 did not count past the channel n-bit mode limit. 1: Channel 4 counted past the channel n-bit mode limit.
13	C3IOVFI	**Channel 3 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 3 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 3 did not count past the channel n-bit mode limit. 1: Channel 3 counted past the channel n-bit mode limit.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 22.8. EPCA0_STATUS Register Bit Descriptions

Bit	Name	Function
12	C2IOVFI	**Channel 2 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 2 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 2 did not count past the channel n-bit mode limit. 1: Channel 2 counted past the channel n-bit mode limit.
11	C1IOVFI	**Channel 1 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 1 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 1 did not count past the channel n-bit mode limit. 1: Channel 1 counted past the channel n-bit mode limit.
10	C0IOVFI	**Channel 0 Intermediate Overflow Interrupt Flag.** This bit is set by hardware when a counter overflows the n-bit range in Channel 0 n-bit PWM mode. This bit must be cleared by firmware. 0: Channel 0 did not count past the channel n-bit mode limit. 1: Channel 0 counted past the channel n-bit mode limit.
9	Reserved	Must write reset value.
8	UPDCF	**Register Update Complete Flag.** 0: An EPCA register update completed or is not pending. 1: An EPCA register update has not completed and is still pending.
7	OVFI	**Counter/Timer Overflow/Limit Interrupt Flag.** This bit is set by hardware when the counter reaches the value in LIMIT and overflows to zero. This bit must be cleared by firmware. 0: An EPCA Counter/Timer overflow/limit event did not occur. 1: An EPCA Counter/Timer overflow/limit event occurred.
6	RUN	**Counter/Timer Run.** 0: Stop the EPCA Counter/Timer. 1: Start the EPCA Counter/Timer.
5	C5CCI	**Channel 5 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 5. This bit must be cleared by firmware. 0: A Channel 5 match or capture event did not occur. 1: A Channel 5 match or capture event occurred.
4	C4CCI	**Channel 4 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 4. This bit must be cleared by firmware. 0: A Channel 4 match or capture event did not occur. 1: A Channel 4 match or capture event occurred.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 22.8. EPCA0_STATUS Register Bit Descriptions

Bit	Name	Function
3	C3CCI	**Channel 3 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 3. This bit must be cleared by firmware. 0: A Channel 3 match or capture event did not occur. 1: A Channel 3 match or capture event occurred.
2	C2CCI	**Channel 2 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 2. This bit must be cleared by firmware. 0: A Channel 2 match or capture event did not occur. 1: A Channel 2 match or capture event occurred.
1	C1CCI	**Channel 1 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 1. This bit must be cleared by firmware. 0: A Channel 1 match or capture event did not occur. 1: A Channel 1 match or capture event occurred.
0	C0CCI	**Channel 0 Capture/Compare Interrupt Flag.** This bit is set by hardware when a match or capture occurs in Channel 0. This bit must be cleared by firmware. 0: A Channel 0 match or capture event did not occur. 1: A Channel 0 match or capture event occurred.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 22.4. EPCA0_COUNTER: Module Counter/Timer

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COUNTER															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
EPCA0_COUNTER = 0x4000_E1B0

Table 22.9. EPCA0_COUNTER Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	COUNTER	**Counter/Timer.** This field is the current value of EPCA counter/timer.

Register 22.5. EPCA0_LIMIT: Module Upper Limit

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LIMIT															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Register ALL Access Address EPCA0_LIMIT = 0x4000_E1C0																

Table 22.10. EPCA0_LIMIT Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	LIMIT	**Upper Limit.** The EPCA Counter/Timer counts from 0 to the upper limit represented by this field.

Register 22.6. EPCA0_LIMITUPD: Module Upper Limit Update Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LIMITUPD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EPCA0_LIMITUPD = 0x4000_E1D0

Table 22.11. EPCA0_LIMITUPD Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	LIMITUPD	**Module Upper Limit Update Value.** This field will be transferred to the LIMIT field when a EPCA counter/timer overflow/limit event occurs if updates are allowed (NOUPD = 0). The UPDCF bit will be set to 1 by hardware when the transfer operation is complete.

Register 22.7. EPCA0_DTIME: Phase Delay Time

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DTIMEY								DTIMEX							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EPCA0_DTIME = 0x4000_E1E0

Table 22.12. EPCA0_DTIME Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:8	DTIMEY	**Y Phase Delay Time.** This field is the amount of time in AHB clock cycles after the differential Y Phase output de-asserts before the X Phase output can assert. This is a global setting for all channels in the EPCA module.
7:0	DTIMEX	**X Phase Delay Time.** This field is the amount of time in AHB clock cycles after the differential X Phase output de-asserts before the Y Phase output can assert. This is a global setting for all channels in the EPCA module.

Register 22.8. EPCA0_DTARGET: DMA Transfer Target

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DTARGET[31:16]															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DTARGET[15:0]															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EPCA0_DTARGET = 0x4000_E200

Table 22.13. EPCA0_DTARGET Register Bit Descriptions

Bit	Name	Function
31:0	DTARGET	**DMA Transfer Target.** Writes to this field will be written to the EPCA register selected by the DPTR field.

Notes:

1. The access methods for this register are restricted. Do not use half-word or byte access methods on this register.

22.10. EPCA0 Register Memory Map

Table 22.14. EPCA0 Memory Map

Register Name	EPCA0_LIMIT	EPCA0_COUNTER	EPCA0_STATUS	EPCA0_CONTROL	EPCA0_MODE	Register Name
ALL Address	0x4000_E1C0	0x4000_E1B0	0x4000_E1A0	0x4000_E190	0x4000_E180	ALL Address
Access Methods	ALL	ALL	ALL | SET | CLR	ALL | SET | CLR	ALL	Access Methods
Bit 31	Reserved	Reserved	Reserved	DIV	Reserved	Bit 31
Bit 30						Bit 30
Bit 29						Bit 29
Bit 28					STDOSEL	Bit 28
Bit 27						Bit 27
Bit 26					Reserved	Bit 26
Bit 25					DBUSYF	Bit 25
Bit 24					DSTART	Bit 24
Bit 23						Bit 23
Bit 22						Bit 22
Bit 21				DIVST	DPTR	Bit 21
Bit 20				Reserved		Bit 20
Bit 19						Bit 19
Bit 18					DEND	Bit 18
Bit 17						Bit 17
Bit 16						Bit 16
Bit 15			C5IOVFI	STEN	Reserved	Bit 15
Bit 14			C4IOVFI	STESEL		Bit 14
Bit 13			C3IOVFI	STSEL	CLKSEL	Bit 13
Bit 12			C2IOVFI			Bit 12
Bit 11			C1IOVFI	Reserved		Bit 11
Bit 10			C0IOVFI			Bit 10
Bit 9			Reserved	DBGMD	CLKDIV	Bit 9
Bit 8			UPDCF	Reserved		Bit 8
Bit 7			OVFI	NOUPD		Bit 7
Bit 6			RUN	Reserved		Bit 6
Bit 5			C5CCI	OVFSEN		Bit 5
Bit 4			C4CCI	OVFDEN		Bit 4
Bit 3			C3CCI	OVFIEN		Bit 3
Bit 2			C2CCI			Bit 2
Bit 1			C1CCI			Bit 1
Bit 0			C0CCI			Bit 0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 22.14. EPCA0 Memory Map

Register Name	EPCA0_DTARGET	EPCA0_DTIME	EPCA0_LIMITUPD
ALL Address	0x4000_E200	0x4000_E1E0	0x4000_E1D0
Access Methods	ALL	ALL	ALL
Bit 31	DTARGET	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15		DTIMEY	LIMITUPD
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9			
Bit 8			
Bit 7		DTIMEX	
Bit 6			
Bit 5			
Bit 4			
Bit 3			
Bit 2			
Bit 1			
Bit 0			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

22.11. EPCA0_CH0-5 Registers

This section contains the detailed register descriptions for EPCA0_CH0, EPCA0_CH1, EPCA0_CH2, EPCA0_CH3, EPCA0_CH4 and EPCA0_CH5 registers.

Register 22.9. EPCAn_CHx_MODE: Channel Capture/Compare Mode

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					CMD			Reserved	DIFGEN	PWMMD				COSEL	
Type	R					RW			R	RW	RW				RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

EPCA0_CH0_MODE = 0x4000_E000
EPCA0_CH1_MODE = 0x4000_E040
EPCA0_CH2_MODE = 0x4000_E080
EPCA0_CH3_MODE = 0x4000_E0C0
EPCA0_CH4_MODE = 0x4000_E100
EPCA0_CH5_MODE = 0x4000_E140

Table 22.15. EPCAn_CHx_MODE Register Bit Descriptions

Bit	Name	Function
31:11	Reserved	Must write reset value.
10:8	CMD	**Channel Operating Mode.** 000: Configure the channel for edge-aligned PWM mode. 001: Configure the channel for center-aligned PWM mode. 010: Configure the channel for high-frequency/square-wave mode. 011: Configure the channel for timer/capture mode. 100: Configure the channel for n-bit edge-aligned PWM mode. 101-111: Reserved.
7	Reserved	Must write reset value.
6	DIFGEN	**Differential Signal Generator Enable.** 0: Disable the differential signal generator. The channel will output a single non-differential output. 1: Enable the differential signal generator. The channel will output two differential outputs: X Phase (XPH) and Y Phase (YPH).

Table 22.15. EPCAn_CHx_MODE Register Bit Descriptions

Bit	Name	Function
5:2	PWMMD	**PWM N-Bit Mode.** This field represents the n-bit PWM for this channel. When in n-bit PWM mode, the channel will behave as if the EPCA Counter/Timer is only n bits wide.
1:0	COSEL	**Channel Output Function Select.** 00: Toggle the channel output at the next capture/compare, overflow, or intermediate event. 01: Set the channel output at the next capture/compare, overflow, or intermediate event. 10: Clear the output at the next capture/compare, overflow, or intermediate event. 11: Capture/Compare, overflow, or intermediate events do not control the output state.

Register 22.10. EPCAn_CHx_CONTROL: Channel Capture/Compare Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved		CIOVFSEN	CIOVFDEN	CIOVFIEN	CCSEN	CCDEN	CCIEN	XPHST	ACTIVEPH	YPHST	Reserved	CUPDCF	CNCAPEN	CPCAPEN	COUTST
Type	R		RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

EPCA0_CH0_CONTROL = 0x4000_E010
EPCA0_CH1_CONTROL = 0x4000_E050
EPCA0_CH2_CONTROL = 0x4000_E090
EPCA0_CH3_CONTROL = 0x4000_E0D0
EPCA0_CH4_CONTROL = 0x4000_E110
EPCA0_CH5_CONTROL = 0x4000_E150
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 22.16. EPCAn_CHx_CONTROL Register Bit Descriptions

Bit	Name	Function
31:14	Reserved	Must write reset value.
13	CIOVFSEN	**Intermediate Overflow Synchronization Signal Enable.** 0: Do not send a synchronization signal when a channel intermediate overflow event occurs. 1: Send a synchronization signal when a channel intermediate overflow occurs.
12	CIOVFDEN	**Intermediate Overflow DMA Request Enable.** 0: Do not request DMA data when a channel intermediate overflow event occurs. 1: Request DMA data when a channel intermediate overflow event occurs.
11	CIOVFIEN	**Intermediate Overflow Interrupt Enable.** 0: Disable the channel intermediate overflow interrupt. 1: Enable the channel intermediate overflow interrupt.
10	CCSEN	**Capture/Compare Synchronization Signal Enable.** 0: Do not send a synchronization signal when a channel capture/compare event occurs. 1: Send a synchronization signal when a channel capture/compare event occurs.

Table 22.16. EPCAn_CHx_CONTROL Register Bit Descriptions

Bit	Name	Function
9	CCDEN	**Capture/Compare DMA Request Enable.** 0: Do not request DMA data when a channel capture/compare event occurs. 1: Request DMA data when a channel capture/compare event occurs.
8	CCIEN	**Capture/Compare Interrupt Enable.** 0: Disable the channel capture/compare interrupt. 1: Enable the channel capture/compare interrupt.
7	XPHST	**Differential X Phase State.** 0: Set the X Phase output state to low. 1: Set the X Phase output state to high.
6	ACTIVEPH	**Active Channel Select.** This bit indicates which phase logic is currently controlling the differential outputs. 0: The Y Phase is active and X Phase is inactive. 1: The X Phase is active and Y Phase is inactive.
5	YPHST	**Differential Y Phase State.** 0: Set the Y Phase output state to low. 1: Set the Y Phase output state to high.
4	Reserved	Must write reset value.
3	CUPDCF	**Channel Register Update Complete Flag.** 0: A EPCA channel register update completed or is not pending. 1: A EPCA channel register update has not completed and is still pending.
2	CNCAPEN	**Negative Edge Input Capture Enable.** 0: Disable negative-edge input capture. 1: Enable negative-edge input capture.
1	CPCAPEN	**Positive Edge Input Capture Enable.** 0: Disable positive-edge input capture. 1: Enable positive-edge input capture.
0	COUTST	**Channel Output State.** 0: The channel output state is low. 1: The channel output state is high.

Register 22.11. EPCAn_CHx_CCAPV: Channel Compare Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved														CCAPV[17:16]	
Type	R														RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CCAPV[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

EPCA0_CH0_CCAPV = 0x4000_E020
EPCA0_CH1_CCAPV = 0x4000_E060
EPCA0_CH2_CCAPV = 0x4000_E0A0
EPCA0_CH3_CCAPV = 0x4000_E0E0
EPCA0_CH4_CCAPV = 0x4000_E120
EPCA0_CH5_CCAPV = 0x4000_E160

Table 22.17. EPCAn_CHx_CCAPV Register Bit Descriptions

Bit	Name	Function
31:18	Reserved	Must write reset value.
17:0	CCAPV	**Channel Compare Value.** This field holds the channel compare value for comparator functions or the channel capture data from timer functions. The LSB represents 1/2 PCA clock period.

Register 22.12. EPCAn_CHx_CCAPVUPD: Channel Compare Update Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved														CCAPVUPD[17:16]	
Type	R														RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CCAPVUPD[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

EPCA0_CH0_CCAPVUPD = 0x4000_E030
EPCA0_CH1_CCAPVUPD = 0x4000_E070
EPCA0_CH2_CCAPVUPD = 0x4000_E0B0
EPCA0_CH3_CCAPVUPD = 0x4000_E0F0
EPCA0_CH4_CCAPVUPD = 0x4000_E130
EPCA0_CH5_CCAPVUPD = 0x4000_E170

Table 22.18. EPCAn_CHx_CCAPVUPD Register Bit Descriptions

Bit	Name	Function
31:18	Reserved	Must write reset value.
17:0	CCAPVUPD	**Channel Compare Update Value.** This field will be transferred to the CCAPV field when a EPCA counter/timer overflow occurs if updates are allowed (NOUPD = 0). The CUPDCF bit will be set to 1 by hardware when the transfer operation is complete.

22.12. EPCAn_CHx Register Memory Map

Table 22.19. EPCAn_CHx Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EPCAn_CHx_MODE	0x0	ALL	Reserved																					CMD			Reserved	DIFGEN	PWMMD				COSEL	
EPCAn_CHx_CONTROL	0x10	ALL | SET | CLR	Reserved																		CIOVFSEN	CIOVFDEN	CIOVFIEN	CCSEN	CCDEN	CCIEN	XPHST	ACTIVEPH	YPHST	Reserved	CUPDCF	CNCAPEN	CPCAPEN	COUTST

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: EPCA0_CH0 = 0x4000_E000, EPCA0_CH1 = 0x4000_E040, EPCA0_CH2 = 0x4000_E080, EPCA0_CH3 = 0x4000_E0C0, EPCA0_CH4 = 0x4000_E100, EPCA0_CH5 = 0x4000_E140

Table 22.19. EPCAn_CHx Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EPCAn_CHx_CCAPV	0x20	ALL	Reserved														CCAPV																		
EPCAn_CHx_CCAPVUPD	0x30	ALL	Reserved														CCAPVUPD																		

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: EPCA0_CH0 = 0x4000_E000, EPCA0_CH1 = 0x4000_E040, EPCA0_CH2 = 0x4000_E080, EPCA0_CH3 = 0x4000_E0C0, EPCA0_CH4 = 0x4000_E100, EPCA0_CH5 = 0x4000_E140

23. External Oscillator (EXTOSC0)

This section describes the External Oscillator (EXTOSC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the EXTOSC block, which is used by all device families covered in this document.

23.1. External Oscillator Features

The External Oscillator control has the following features:

- Support for external crystal or ceramic resonator, RC, C, or CMOS oscillators.
- Support external CMOS frequencies from 10 kHz to 50 MHz and external crystal frequencies from 10 kHz to 25 MHz.
- Various drive strengths for flexible crystal oscillator support.
- Internal frequency divide-by-two option available for some oscillator modes.
- Available as an input to the PLL.

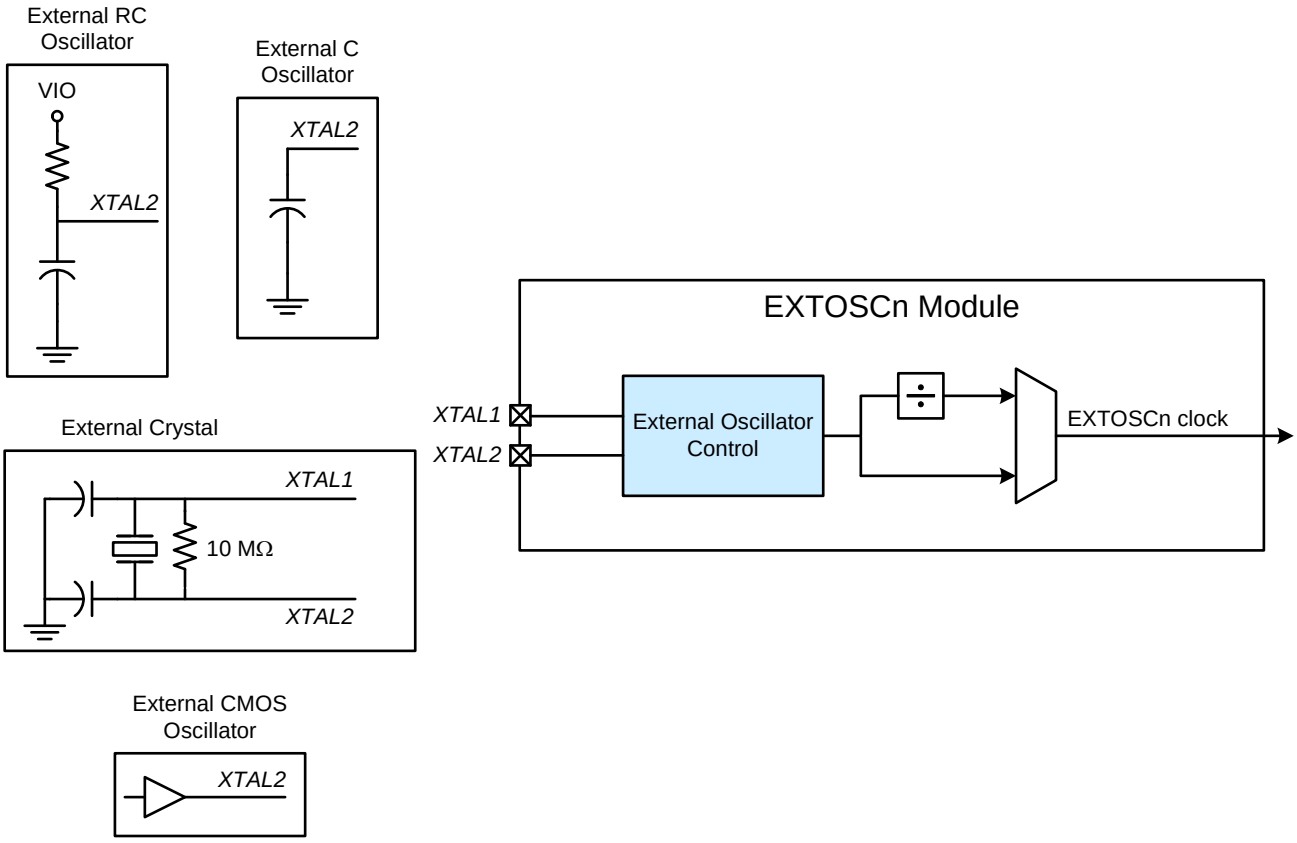


Figure 23.1. External Oscillator Block Diagram

23.2. Introduction

The EXTOSC external oscillator circuit may drive an external crystal, ceramic resonator, capacitor, or RC network. A CMOS clock may also provide a clock input. The external oscillator output may be selected as the AHB clock or used to clock other modules independent of the AHB clock selection.

23.3. External Crystal Oscillator

When OSCMD is 6 or 7, the EXTOSC module is in crystal oscillator mode. The crystal or ceramic resonator and a 10 MΩ resistor must be wired across the XTAL1 and XTAL2 pins as shown in Figure 23.2. Appropriate loading capacitors should be added to XTAL1 and XTAL2, and both pins should be configured for analog mode as described in the port configuration module.

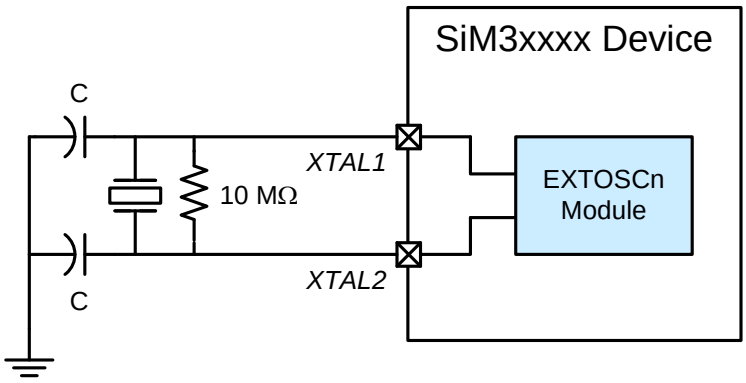


Figure 23.2. External Crystal Oscillator Configuration

The capacitors provide the load capacitance required by the crystal for correct oscillation. These capacitors are “in series” as seen by the crystal and “in parallel” with the stray capacitance of the XTAL1 and XTAL2 pins.

Note: The recommended load capacitance depends upon the crystal and the manufacturer. Refer to the crystal data sheet when completing these calculations.

Equation 23.1 describes the equation for determining the load capacitance for the two capacitors. The C_A and C_B values are the capacitors connected to the crystal leads. The C_S value is the total stray capacitance of the PCB.

$$C_L = \frac{C_A \times C_B}{C_A + C_B} + C_S$$

Equation 23.1. Crystal Load Capacitors

If C_A and C_B are the same (C), the resulting equation is shown in Equation 23.2.

$$C_L = \frac{C}{2} + C_S$$

Equation 23.2. Simplified Crystal Load Capacitors

For example, using Equation 23.2 with a 32.768 kHz tuning-fork crystal with a recommended load capacitance of 12.5 pF placed as close to the pins as possible (assuming 6 pF total stray capacitance) results in crystal load capacitors of 13 pF each.

Note: Crystal oscillator circuits are quite sensitive to PCB layout. The crystal should be placed as close as possible to the XTAL pins on the device. The traces should be as short as possible and shielded with ground plane from any other traces that could introduce noise or interference.

Setting OSCMD to 6 places EXTOSC in crystal oscillator mode, and setting OSCMD to 7 sets the module in crystal oscillator divided by 2 mode. This divide-by-2 stage ensures that the clock derived from the external oscillator has a duty cycle of 50%. When operating in either crystal mode, the frequency control (FREQCN) field must be set to the appropriate value based on the crystal frequency.

Table 23.1. Frequency Control for Crystal Oscillators

Crystal Frequency	FREQCN Value
10 kHz < f ≤ 20 kHz	0
20 kHz < f ≤ 58 kHz	1
58 kHz < f ≤ 155 kHz	2
155 kHz < f ≤ 415 kHz	3
415 kHz < f ≤ 1.1 MHz	4
1.1 MHz < f ≤ 3.1 MHz	5
3.1 MHz < f ≤ 8.2 MHz	6
8.2 MHz < f ≤ 25 MHz	7

23.3.1. Configuring for Crystal Oscillator Mode

The recommended procedure for starting the crystal is as follows:

1. Configure the XTAL1 and XTAL2 pins for analog mode using the device port configuration module.
2. Disable the XTAL1 and XTAL2 digital output drivers by writing 1's to the pin latch in the Port Bank registers.
3. Configure the FREQCN field for the crystal frequency according to Table 23.1.
4. Set the OSCMD field to 6 for crystal mode or 7 for crystal divided by 2 mode.
5. Wait at least 1 ms.
6. Poll on the OSCVLDF flag to determine if the oscillator is running and stable.
7. Set the EXTOSCEN bit in CLKCTRL0_CONFIG to 1, to enable the external oscillator as a clock source.
8. Select the external oscillator as the AHB clock or as the input clock for a module.

23.4. External CMOS Oscillator

If an external CMOS clock is used as the external oscillator, the clock should be directly routed into XTAL2, and the XTAL2 pin should be configured as a digital input using the device port configuration module. XTAL1 is not used in external CMOS clock mode.

The CMOS oscillator mode is available with a divide by 2 stage, which ensures that the clock derived from the external oscillator has a duty cycle of 50%.

The external oscillator valid (OSCVLDF) flag will always return zero when the external oscillator is configured to external CMOS clock mode.

23.4.1. Configuring for CMOS Oscillator Mode

The recommended procedure for enabling the CMOS oscillator is:

1. Configure the XTAL2 pins for digital input mode using the device port configuration module.
2. Set the OSCMD field to 2 for CMOS mode or 3 for CMOS mode with divide by 2 stage.
3. Set the EXTOSCEN bit in CLKCTRL0_CONFIG to 1, to enable the external oscillator as a clock source.
4. Select the external oscillator as the AHB clock or as the input clock for a module.

23.5. External RC Oscillator

When using the EXTOSC module with an external RC network, firmware should set the OSCMD field to 4 for RC divided by 2 mode. The RC network should be added to XTAL2, and XTAL2 should be configured for analog mode with the digital output drivers disabled. XTAL1 is not affected in RC mode. Figure 23.3 shows this hardware configuration.

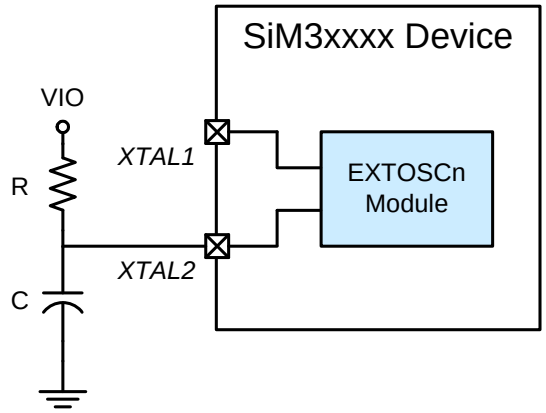


Figure 23.3. External RC Oscillator Configuration

The capacitor used in the RC network should have a value no greater than 100 pF, and the resistor should be no smaller than 10 kΩ. For very small capacitors, the parasitic capacitance in the PCB layout may dominate the total capacitance. The oscillation frequency can be determined by Equation 23.3, where F is the frequency in MHz, R is the pull-up resistor value in kΩ., and C is the capacitor value in the XTAL2 pin in pF.

$$F = \frac{1.23 \times 10^3}{R \times C}$$

Equation 23.3. RC Oscillation Frequency

To determine the required frequency control (FREQCN), first select the RC network value to produce the desired frequency of oscillation. For example, if the desired frequency is 100 kHz, let R = 246 kΩ. and C = 50 pF.

$$F = \frac{1.23 \times 10^3}{R \times C} = \frac{1.23 \times 10^3}{246 \times 50} = 100\ \text{kHz}$$

Table 23.2 shows the frequency ranges for the frequency control (FREQCN) bit in RC oscillator mode. The recommended FREQCN setting for this example is 2.

The RC oscillator mode is only available with a divide by 2 stage, which ensures that the clock derived from the external oscillator has a duty cycle of 50%. The equation for the EXTOSC output frequency is shown in Equation 23.4.

$$F_{OUT} = \frac{F}{2} = \frac{1.23 \times 10^3}{2 \times R \times C}$$

Equation 23.4. EXTOSC Output Frequency in RC Mode

Table 23.2. Frequency Control for RC Oscillators

RC Oscillator Frequency	EXTOSC Output Frequency (f_{OUT})	FREQCN Value
f ≤ 25 kHz	f_{OUT} ≤ 12.5 kHz	0
25 kHz < f ≤ 50 kHz	12.5 kHz < f_{OUT} ≤ 25 kHz	1
50 kHz < f ≤ 100 kHz	25 kHz < f_{OUT} ≤ 50 kHz	2
100 kHz < f ≤ 200 kHz	50 kHz < f_{OUT} ≤ 100 kHz	3
200 kHz < f ≤ 400 kHz	100 kHz < f_{OUT} ≤ 200 kHz	4
400 kHz < f ≤ 800 kHz	200 kHz < f_{OUT} ≤ 400 kHz	5
800 kHz < f ≤ 1.6 MHz	400 kHz < f_{OUT} ≤ 800 kHz	6
1.6 MHz < f ≤ 3.2 MHz	800 kHz < f_{OUT} ≤ 1.6 MHz	7

23.5.1. Configuring for RC Oscillator Mode

The recommended procedure for enabling the RC oscillator is:

1. Configure the XTAL2 pin for analog mode using the device port configuration module.
2. Disable the XTAL2 digital output driver by writing 1 to the pin latch in the Port Bank registers.
3. Configure the FREQCN field for the RC frequency according to Table 23.2.
4. Set the OSCMD field to 4 for RC divided by 2 mode.
5. Wait at least 1 ms.
6. Poll on the OSCVLDF flag to determine if the oscillator is running and stable.
7. Set the EXTOSCEN bit in CLKCTRL0_CONFIG to 1, to enable the external oscillator as a clock source.
8. Select the external oscillator as the AHB clock or as the input clock for a module.

23.6. External C Oscillator

Firmware can enable the external C oscillator by setting OSCMD to 5 for C oscillator divided by 2 mode. The capacitor used should be no greater than 100 pF, and the parasitic capacitance in the PCB layout may dominate very small capacitors. The hardware configuration for the external C oscillator is shown in Figure 23.4.

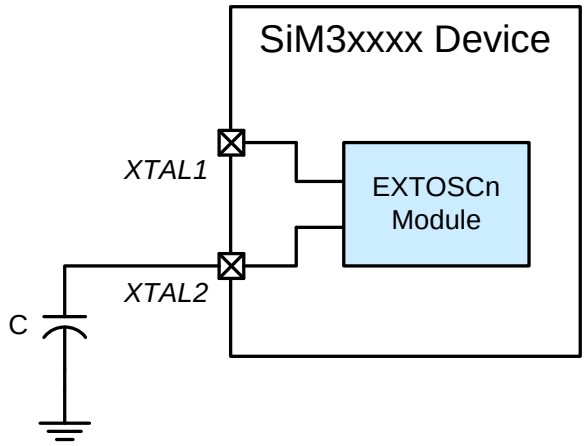


Figure 23.4. External C Oscillator Configuration

To determine the required frequency control (FREQCN) value (XFCN), select the capacitor to be used and find the frequency of oscillation according to Equation 23.5, where F is the frequency of oscillation in MHz, C is the capacitor value in pF, VBAT is the device power supply in Volts, and K is the K Factor.

$$F = \frac{K}{C \times V_{BAT}}$$

Equation 23.5. C Oscillation Frequency

For example, assume VBAT is 3.0 V and the desired C oscillator frequency is 150 kHz. Since the frequency desired is roughly 150 kHz, select the K Factor of 22 from Table 23.3.

$$0.150 = \frac{22}{C \times 3.0}$$

$$C = \frac{22}{0.150 \times 3.0} = 48.8 \text{ pF}$$

Therefore, the FREQCN value to use in this example is 3, and the external capacitor should be 50 pF.

The C oscillator mode is only available with a divide by 2 stage, which ensures that the clock derived from the external oscillator has a duty cycle of 50%. The equation for the EXTOSC output frequency is shown in Equation 23.6.

$$f_{OUT} = \frac{f}{2} = \frac{KF}{2 \times C \times V_{BAT}}$$

Equation 23.6. EXTOSC Output Frequency in C Mode

Table 23.3. Frequency Control for C Oscillators

C Oscillator Frequency	EXTOSC Output Frequency (f_{OUT})	K Factor (KF)	FREQCN Value
f ≤ 25 kHz	f_{OUT} ≤ 12.5 kHz	0.87	0
25 kHz < f ≤ 50 kHz	12.5 kHz < f_{OUT} ≤ 25 kHz	2.6	1
50 kHz < f ≤ 100 kHz	25 kHz < f_{OUT} ≤ 50 kHz	7.7	2
100 kHz < f ≤ 200 kHz	50 kHz < f_{OUT} ≤ 100 kHz	22	3
200 kHz < f ≤ 400 kHz	100 kHz < f_{OUT} ≤ 200 kHz	65	4
400 kHz < f ≤ 800 kHz	200 kHz < f_{OUT} ≤ 400 kHz	180	5
800 kHz < f ≤ 1.6 MHz	400 kHz < f_{OUT} ≤ 800 kHz	664	6
1.6 MHz < f ≤ 3.2 MHz	800 kHz < f_{OUT} ≤ 1.6 MHz	1590	7

23.6.1. Configuring for C Oscillator Mode

The recommended procedure for enabling the C oscillator is as follows:

1. Configure the XTAL2 pin for analog mode using the device port configuration module.
2. Disable the XTAL2 digital output driver by writing 1 to the pin latch in the Port Bank registers.
3. Configure the FREQCN field for the C frequency and K Factor according to Table 23.3.
4. Set the OSCMD field to 5 for C oscillator divided by 2 mode.
5. Wait at least 1 ms.
6. Poll on the OSCVLDF flag to determine if the oscillator is running and stable.
7. Set the EXTOSCEN bit in CLKCTRL0_CONFIG to 1, to enable the external oscillator as a clock source.
8. Select the external oscillator as the AHB clock or as the input clock for a module.

23.7. EXTOSC0 Registers

This section contains the detailed register descriptions for EXTOSC0 registers.

Register 23.1. EXTOSC0_CONTROL: Oscillator Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									OSCMD			OSCVLDF	FREQCN		
Type	R									RW			R	RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

EXTOSC0_CONTROL = 0x4003_C000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 23.4. EXTOSC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:7	Reserved	Must write reset value.
6:4	OSCMD	**Oscillator Mode.** 000: External oscillator off. 001: Reserved. 010: External CMOS clock mode. 011: External CMOS with divide by 2 stage. 100: RC oscillator mode with divide by 2 stage. 101: C oscillator mode with divide by 2 stage. 110: Crystal oscillator mode. 111: Crystal oscillator mode with divide by 2 stage.
3	OSCVLDF	**Oscillator Valid Flag.** This bit indicates when the oscillator has stabilized after an initial startup condition. Hardware will clear the bit to 0 when the external oscillator is disabled, and set the bit to 1 once the oscillator is running properly. It will not clear automatically if oscillation fails. This bit is valid for all modes of operation except external CMOS clock and external CMOS clock divide by 2 modes, when OSCVLDF always reads back as 0. 0: The external oscillator is unused or not yet stable. 1: The external oscillator is running and stable.

Table 23.4. EXTOSC0_CONTROL Register Bit Descriptions

Bit	Name	Function
2:0	FREQCN	**Frequency Control.** 000: Set the external oscillator to range 0. 001: Set the external oscillator to range 1. 010: Set the external oscillator to range 2. 011: Set the external oscillator to range 3. 100: Set the external oscillator to range 4. 101: Set the external oscillator to range 5. 110: Set the external oscillator to range 6. 111: Set the external oscillator to range 7.

23.8. EXTOSC0 Register Memory Map

Table 23.5. EXTOSC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
EXTOSC0_CONTROL	0x4003_C000	ALL | SET | CLR	Reserved																									OSCMD			OSCVLDF	FREQCN		

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

24. Flash Controller (FLASHCTRL0)

This section describes the Flash Controller (FLASHCTRL) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the FLASHCTRL block, which is used by all device families covered in this document.

24.1. Flash Controller Features

The Flash Controller module includes the following features:

- Provides control for read timing and prefetch.
- Provides an access port for firmware to write and erase flash in system.
- Buffers multiple writes to the flash and stalls the core if the buffer is full until the flash operations can complete.
- Secures the flash with a lock and key mechanism.
- Allows single writes and erases or multiple writes with a single unlock operation.
- Blocks modifications to flash if the Supply Monitor is not enabled as a reset source.

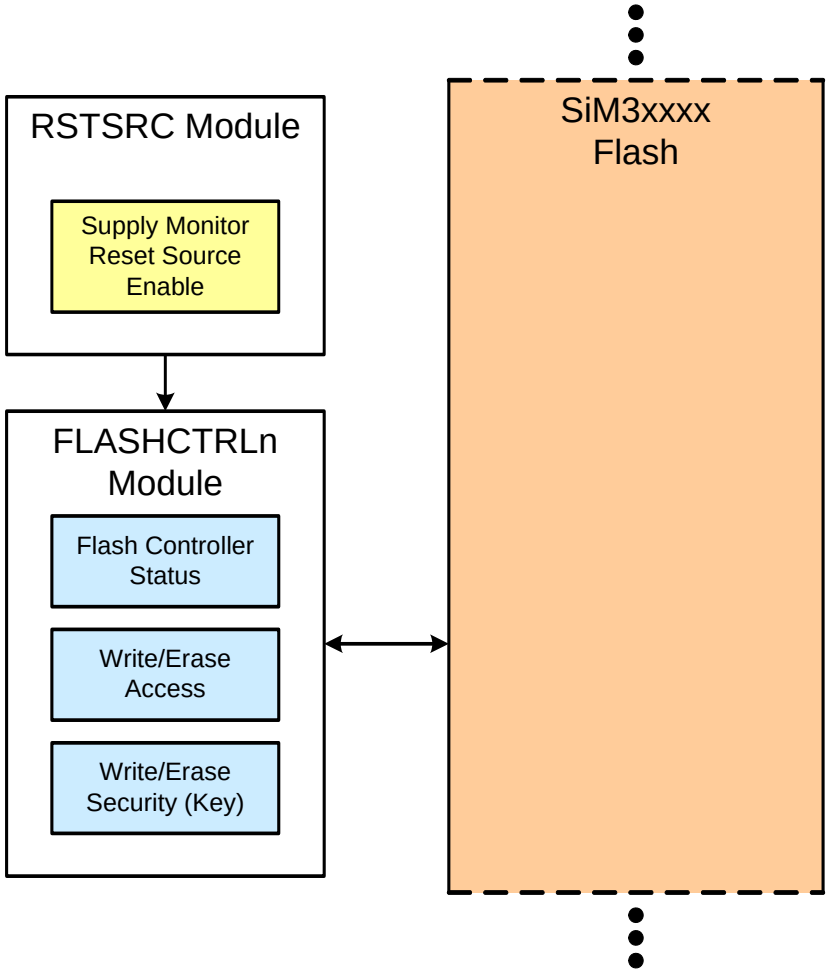


Figure 24.1. Flash Controller Block Diagram

24.2. Overview

The Flash Controller (FLASHCTRL) module provides flash read, write, and erase control. The read controller includes the prefetch engine controls and the flash read timing settings that must be adjusted appropriately for the AHB clock of the device. The write and erase control can be used to modify the flash contents in system. This write and erase interface is protected by a lock and key mechanism to prevent any undesired modification of flash.

24.3. Flash Read Control

The flash read timing is controlled by the speed mode (SPMD) field and the read-store enable (RDSEN) bit. These bits must be set appropriately for the selected AHB frequency for the system.

When read-store enable (RDSEN) is set to 1, the first flash access is stored in the read store pipeline buffer before being passed to the AHB. Otherwise, the first flash access is passed directly to the AHB (read through).

Table 24.1. Read Timing Ranges

SPMD Wait Value	RDSEN Value	Setting	Max AHB Frequency
0	0	Non-pipelined, Latency = 0	21 MHz
0	1	Pipelined, Latency = 0	26 MHz
1	0	Non-pipelined, Latency = 1	43 MHz
1	1	Pipelined, Latency = 1	53 MHz*
2	0	Non-pipelined, Latency = 2	65 MHz*
2	1	Pipelined, Latency = 2	80 MHz*
3	0	Non-pipelined, Latency = 3	87 MHz*
3	1	Pipelined, Latency = 3	107 MHz*

***Note:** Device operation beyond the maximum frequency listed in the data sheet is not guaranteed, and should be avoided.

The flash read timing mode (FLRTMD) can also be configured to save power at slower clock frequencies. It should be set to 1 for AHB clock frequencies above 12 MHz and cleared to 0 for AHB clock frequencies below 12 MHz. The FLRTMD bit should only be set to 1 when RDSEN is disabled and the SPMD Wait value is zero. FLRTMD has no effect on the flash read timing.

The flash read controller also includes prefetch engine controls. Enabling the data prefetch (DPFEN = 1) feature allows data accesses to be stored and queued into the prefetch buffer in addition to instructions, which can help performance if a large amount of data is accessed sequentially in flash. Firmware can also disable the prefetch engine (PFINH = 1), which will reduce performance but improve the device's power consumption.

24.4. Flash Write and Erase Control

The flash write and erase interface allows firmware to modify the flash in system. This interface is protected by a security lock and key interface to prevent inadvertent modifications of memory.

Firmware cannot modify flash through the interface when the supply monitor in the VMONn module is disabled or disabled as a reset source (device reset source module). Any write or erase operations initiated while the supply monitor is disabled or disabled as a reset source will be ignored.

Firmware should disable interrupts when using the interface to modify flash contents. This will ensure the Flash interface accesses are sequential in time and take the minimum time possible.

24.4.1. Security Interface

The flash interface is initially locked after a reset. The interface is unlocked by writing the initial unlock key (0xA5) followed by one of the command keys in consecutive writes to the KEY field. Any writes to the WRDATA register while the interface is locked or an incorrect unlock sequence will permanently lock the Flash interface until the next device reset.

The single unlock key unlocks the flash interface for one write or erase operation. The flash interface will remain unlocked after a single unlock command if firmware writes an additional value to the KEY field before writing to the WRDATA register.

The multiple unlock key unlocks the flash interface for write or erase operations until the multiple lock key is written to the KEY field. The flash interface will remain unlocked if firmware writes any value other than the multiple write lock to KEY. The multiple lock is not a permanent lock, and the interface can be unlocked again for other operations.

Table 24.2. Flash Interface Keys

Command	KEY Value
Initial Unlock	0xA5
Single Unlock	0xF1
Multiple Unlock	0xF2
Multiple Lock	0x5A

24.4.2. Writing and Erasing Flash

Once the flash security interface is unlocked, write and erase operations occur using the write address (WRADDR) and write data (WRDATA) indirect registers. Writes to WRDATA must be half-word aligned, and each half word may only be written once between erase operations. For a write operation (ERASEEN = 0), the right-justified, half-word value written to WRDATA will be written to the address specified by WRADDR. For an erase operation (ERASEEN = 1), a write to WRDATA will initiate an erase on the flash page specified by the WRADDR field. Flash pages are 1024 bytes, and aligned at 1024-byte boundaries in the device, beginning at address 0x0000.

Firmware should write the data to the WRDATA register 16 bits at a time in right-justified format, and hardware automatically increments the WRADDR field by two after each write operation. The data written to the WRDATA register is first placed in the controller write buffer. When writing multiple bytes and executing from RAM, firmware can poll on the BUFSTS flag to wait until the buffer has room before writing more data to the WRDATA register. This allows firmware to perform other actions while the controller is modifying flash. If the buffer is full and firmware writes another half-word to WRDATA, the flash controller will stall the AHB bus until the write operation completes, and the buffer is no longer full. Using this method, firmware can write to WRDATA in a series of successive writes without having to poll.

Note: For all flash write operations, firmware will stall unless operating from a memory space other than Flash.

The flash controller can write multiple sequential half-words to flash faster than using individual accesses if the Flash interface is unlocked for multiple byte writes and sequential writes are enabled (SQWEN = 1). Firmware using this feature should run from a memory space other than flash. Otherwise, the flash controller switches out of sequential mode for the Flash read and back into sequential mode for the write, which causes a longer delay than individual accesses with SQWEN cleared to 0.

The busy (BUSYF) flag indicates when the flash controller is currently executing a Flash write or erase operation.

24.4.3. Writing a Single Half-Word to Flash

To write a single byte to flash:

1. Ensure the supply monitor is enabled and enabled as a reset source in the RSTSRC module.
2. Disable erase operations (ERASEEN = 0).
3. Write the destination address to WRADDR.
4. Disable interrupts.
5. Write the initial unlock value to KEY (0xA5).
6. Write the single unlock value to KEY (0xF1).
7. Write the data half-word to WRDATA in right-justified format.
8. (Optional) If executing code from a memory space other than Flash, poll on the BUSYF flag until hardware clears it to 0.
9. Enable interrupts.

24.4.4. Writing Multiple Half-Words to Sequential Flash Addresses

To write a sequential set of bytes to flash, code should execute from a memory space other than flash and complete the following steps:

1. Ensure the supply monitor is enabled and enabled as a reset source in the RSTSRC module.
2. Disable erase operations (ERASEEN = 0).
3. Write the initial destination address to WRADDR.
4. Enable sequential writes (SQWEN = 1).
5. Disable interrupts.
6. Write the initial unlock value to KEY (0xA5).
7. Write the multiple unlock value to KEY (0xF2).
8. Write the data half-word to WRDATA in right-justified format.
9. (Optional) Poll on the BUFSTS flag until the buffer has room for more data. If code is executing from RAM, this allows the core to perform other actions until a write operation completes and the buffer has room. The AHB bus will automatically stall until the operation completes if firmware writes data to WRDATA when the buffer is full.
10. Repeat steps 8 and 9 until all data is written. Hardware automatically increments the WRADDR field by 2 after each write operation.
11. (Optional) If executing code from a memory space other than Flash, poll on the BUSYF flag until hardware clears it to 0.
12. Write the multiple lock value to KEY (0x5A).
13. Enable interrupts.

24.4.5. Writing Multiple Half-Words to Non-Sequential Flash Addresses

To write multiple bytes to non-sequential addresses in flash:

1. Ensure the supply monitor is enabled and enabled as a reset source (device reset sources module).
2. Disable erase operations (ERASEEN = 0).
3. Disable interrupts.
4. Write the initial unlock value to KEY (0xA5).

5. Write the multiple unlock value to KEY (0xF2).
6. Write the destination address to WRADDR.
7. Write the data half-word to WRDATA in right-justified format.
8. (Optional) If executing code from a memory space other than flash, poll on the BUSYF flag until hardware clears it to 0.
9. Repeat steps 6, 7, and 8 until all data is written.
10. Write the multiple lock value to KEY (0x5A).
11. Enable interrupts.

24.4.6. Erasing a Page of Flash

To erase a page of Flash:

1. Ensure the supply monitor is enabled and enabled as a reset source (device reset sources module).
2. Write the address of a byte in the Flash page to WRADDR.
3. Enable erase operations (ERASEEN = 1).
4. Disable interrupts.
5. Write the initial unlock value to KEY (0xA5).
6. Write the single unlock value to KEY (0xF1).
7. Write any value to WRDATA in right-justified format to initiate the page erase.
8. (Optional) If executing code from a memory space other than Flash, poll on the BUSYF flag until hardware clears it to 0.
9. Enable interrupts.

24.4.7. Erasing Multiple Flash Pages

To erase multiple pages of flash:

1. Ensure the supply monitor is enabled and enabled as a reset source (device reset sources module).
2. Enable erase operations (ERASEEN = 1).
3. Disable interrupts.
4. Write the initial unlock value to KEY (0xA5).
5. Write the multiple unlock value to KEY (0xF2).
6. Write the address of a byte in the Flash page to WRADDR.
7. Write any value to WRDATA in right-justified format to initiate the page erase.
8. (Optional) If executing code from a memory space other than Flash, poll on the BUSYF flag until hardware clears it to 0.
9. Repeat steps 6, 7, and 8 for each page.
10. Write the multiple lock value to KEY (0x5A).
11. Enable interrupts.

24.5. FLASHCTRL0 Registers

This section contains the detailed register descriptions for FLASHCTRL0 registers.

Register 24.1. FLASHCTRL0_CONFIG: Controller Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved											BUSYF	BUFSTS	ERASEEN	Reserved	SQWEN
Type	R											R	R	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								PFINH	DPFEN	Reserved	RDSEN	Reserved		SPMD	
Type	R					RW			RW	RW	RW	RW	R		RW	
Reset	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0

Register ALL Access Address

FLASHCTRL0_CONFIG = 0x4002_E000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 24.3. FLASHCTRL0_CONFIG Register Bit Descriptions

Bit	Name	Function
31:21	Reserved	Must write reset value.
20	BUSYF	**Flash Operation Busy Flag.** 0: The flash interface is not busy. 1: The flash interface is busy with an operation.
19	BUFSTS	**Flash Buffer Status.** 0: The flash controller write data buffer is empty. 1: The flash controller write data buffer is full.
18	ERASEEN	**Flash Page Erase Enable.** 0: Writes to the WRDATA field will initiate a write to flash at the address in the WRADDR field. 1: Writes to the WRDATA field will initiate an erase of the flash page containing the address in the WRADDR field.
17	Reserved	Must write reset value.

Table 24.3. FLASHCTRL0_CONFIG Register Bit Descriptions

Bit	Name	Function
16	SQWEN	**Flash Write Sequence Enable.** Setting this bit will cause multiple sequential accesses to the flash interface to take less time than individual accesses if the flash interface is unlocked for multiple byte writes. When this bit is used, firmware should be running from a memory space other than flash. 0: Disable sequential write mode. 1: Enable sequential write mode.
15:8	Reserved	Must write reset value.
7	PFINH	**Prefetch Inhibit.** Disabling the prefetch can cause slower performance and better power consumption. 0: Any reads from flash are prefetched until the prefetch buffer is full. 1: Inhibit the prefetch engine.
6	DPFEN	**Data Prefetch Enable.** Setting this bit allows data accesses to be stored and queued into the prefetch buffer, which can help performance if a large amount of data is accessed sequentially in flash. 0: Data accesses are excluded from the prefetch buffer. 1: Data accesses are included in the prefetch buffer.
5	Reserved	Must write reset value.
4	RDSEN	**Read Store Mode Enable.** When set to 1, the first flash access is stored in the prefetch (read store) before being passed to the AHB. Otherwise, the first flash access is passed directly to the AHB (read through). 0: Disable read store mode. 1: Enable read store mode.
3:2	Reserved	Must write reset value.
1:0	SPMD	**Flash Speed Mode.** The flash speed mode must be adjusted appropriately for the system AHB frequency. 00: Read and write the flash at speed mode 0. 01: Read and write the flash at speed mode 1. 10: Read and write the flash at speed mode 2. 11: Read and write the flash at speed mode 3.

Register 24.2. FLASHCTRL0_WRADDR: Flash Write Address

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WRADDR[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	WRADDR[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

FLASHCTRL0_WRADDR = 0x4002_E0A0

Table 24.4. FLASHCTRL0_WRADDR Register Bit Descriptions

Bit	Name	Function
31:0	WRADDR	**Flash Write Address.** The flash operation will occur at the address (write) or page (erase) specified by this field.

Register 24.3. FLASHCTRL0_WRDATA: Flash Write Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WRDATA[31:16]															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	WRDATA[15:0]															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

FLASHCTRL0_WRDATA = 0x4002_E0B0

Table 24.5. FLASHCTRL0_WRDATA Register Bit Descriptions

Bit	Name	Function
31:0	WRDATA	**Flash Write Data.** When erases are enabled, a write to this field will initiate an erase of the flash page containing the address specified by WRADDR. When erases are disabled, a right-justified, half-word write to this field will write the value to the flash address specified by WRADDR. Any data written to the upper half of WRDATA is ignored.

Register 24.4. FLASHCTRL0_KEY: Flash Modification Key

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								KEY							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

FLASHCTRL0_KEY = 0x4002_E0C0

Table 24.6. FLASHCTRL0_KEY Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:0	KEY	**Flash Key.** Writing the initial unlock key (0xA5) followed by the single unlock key (0xF1) to this field will unlock the flash interface for single write or erase operations. The interface will relock after the operation. Writing the initial unlock key (0xA5) followed by the multiple unlock key (0xF2) will unlock the flash interface for multiple write and erase operations. The interface will remain unlocked until the multiple lock key (0x5A) is written to KEY. All other values for the KEY field are reserved.

Register 24.5. FLASHCTRL0_TCONTROL: Flash Timing Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R														RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									FLRTMD	Reserved					
Type	R									RW	RW					
Reset	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	0

Register ALL Access Address

FLASHCTRL0_TCONTROL = 0x4002_E0D0

Table 24.7. FLASHCTRL0_TCONTROL Register Bit Descriptions

Bit	Name	Function
31:7	Reserved	Must write reset value.
6	FLRTMD	**Flash Read Timing Mode.** 0: Configure the flash read controller for AHB clocks below 12 MHz. 1: Configure the flash read controller for AHB clocks above 12 MHz.
5:0	Reserved	Must write reset value.

24.6. FLASHCTRL0 Register Memory Map

Table 24.8. FLASHCTRL0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FLASHCTRL0_CONFIG	0x4002_E000	ALL | SET | CLR	Reserved											BUSYF	BUFSTS	ERASEEN	Reserved	SQWEN	Reserved								PFINH	DPFEN	Reserved	RDSEN	Reserved		SPMD	
FLASHCTRL0_WRADDR	0x4002_E0A0	ALL	WRADDR																															
FLASHCTRL0_WRDATA	0x4002_E0B0	ALL	WRDATA																															

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 24.8. FLASHCTRL0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FLASHCTRL0_KEY	0x4002_E0C0	ALL	Reserved																								KEY							
FLASHCTRL0_TCONTROL	0x4002_E0D0	ALL	Reserved																									FLRTMD	Reserved					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

25. Inter-Integrated Circuit Bus (I2C0)

This section describes the I2C module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the I2C block, which is used by I2C0 on all device families covered in this document.

25.1. I2C Features

The I2C module includes the following features:

- Standard (up to 100 kbps) and Fast (400 kbps) transfer speeds.
- Can operate down to APB clock divided by 32768 or up to APB clock divided by 8.
- Support for master, slave, and multi-master modes.
- Hardware synchronization and arbitration for multi-master mode.
- Clock low extending (clock stretching) to interface with faster masters.
- Hardware support for 7-bit slave and general call address recognition.
- Firmware support for 10-bit slave address decoding.
- Ability to disable all slave states.
- Programmable clock high and low period.
- Programmable data setup/hold times.
- Spike suppression up to 2 times the APB period.

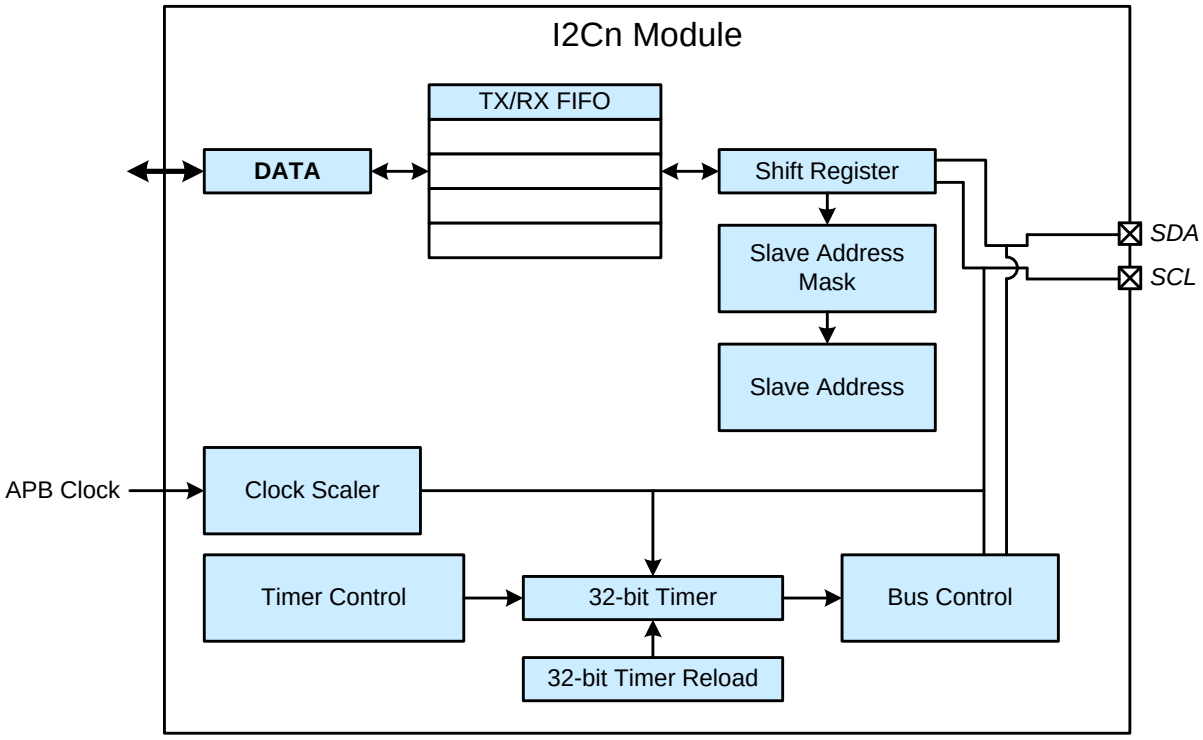


Figure 25.1. I2C Block Diagram

25.2. I2C Protocol

The I2C interface is a two-wire, bi-directional serial bus. Reads and writes to the interface are byte oriented with the I2C interface autonomously controlling the serial transfer of the data. Data can be transferred at up to 1/8th of the APB clock as a master or slave (this can be faster than allowed by the I2C specification, depending on the clock source used). A method of extending the clock-low duration is available to accommodate devices with different speed capabilities on the same bus.

The I2C interface may operate as a master and/or slave, and may function on a bus with multiple masters. The I2C provides control of SDA (serial data), SCL (serial clock) generation and synchronization, arbitration logic, and start/stop control and generation.

25.2.1. Hardware Configuration

Figure 25.2 shows a typical I2C configuration. The I2C specification allows any recessive voltage between 3.0 and 5.0 V; different devices on the bus may operate at different voltage levels. The bi-directional SCL (serial clock) and SDA (serial data) lines must be connected to a positive power supply voltage through a pull-up resistor or similar circuit. Every device connected to the bus must have an open-drain or open-collector output for both the SCL and SDA lines, so that both are pulled high (recessive state) when the bus is free. The maximum number of devices on the bus is limited only by the requirement that the rise and fall times on the bus not exceed 1000 ns (rise) and 300 ns (fall) for Standard mode communication and 300 ns (rise) and 300 ns (fall) for fast mode communication.

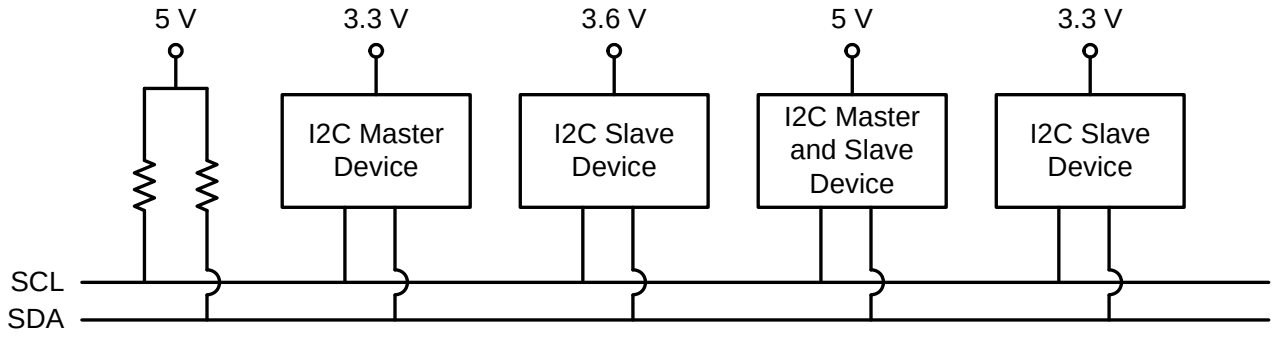


Figure 25.2. I2C Hardware Configuration

A typical I2C transaction consists of a start condition followed by an address (7- or 10-bit), the Read/Write direction bit, one or more bytes of data, and a stop condition. Address and data fields are transferred MSB first. Each byte that is received (by a master or slave) must be acknowledged (ACK) with a low SDA during a high SCL. If the receiving device does not ACK, the transmitting device will read a NACK (not acknowledge), which is a high SDA during a high SCL.

The direction bit (R/W) occupies the least-significant bit position of the address. The direction bit is set to 1 to indicate a read operation and cleared to 0 to indicate a write operation.

A start condition occurs when the SDA signal changes from high to low while SCL is high, and a stop condition occurs when the SDA signal changes from low to high while SCL is high. During normal data bit transitions, the SCL signal is low while SDA changes state.

All transactions are initiated by a master, with one or more addressed slave devices as the target. The master generates the start condition and then transmits the slave address and direction bit. If the transaction is a write operation from the master to the slave, the master transmits the data a byte at a time and waits for an ACK from the slave at the end of each byte. For read operations, the slave transmits the data and waits for an ACK from the master at the end of each byte. At the end of the data transfer, the master generates a stop condition to terminate the transaction and free the bus. Figure 25.3 illustrates a typical I2C 7-bit address transaction.

Both address sizes (7- and 10-bit) can be used on the same I2C bus.

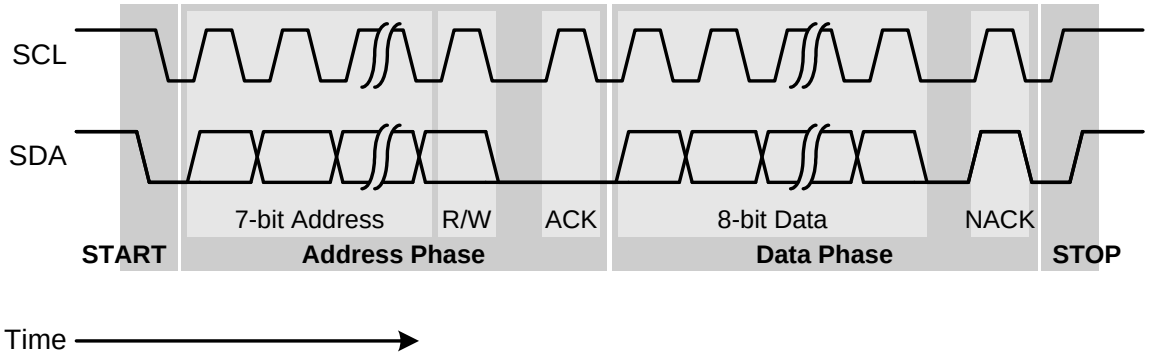


Figure 25.3. Typical I2C 7-bit Address Transaction

25.2.2. Address Phase

The address phase consists of the 7- or 10-bit address, the read/write direction bit (R/W), and the slave's ACK or NACK response.

With 7-bit address transfers, the address phase consists of a single byte operation and one ACK or NACK response from the slave. The R/W bit is the last bit transmitted before the ACK or NACK. Figure 25.4 shows the 7-bit address phase.

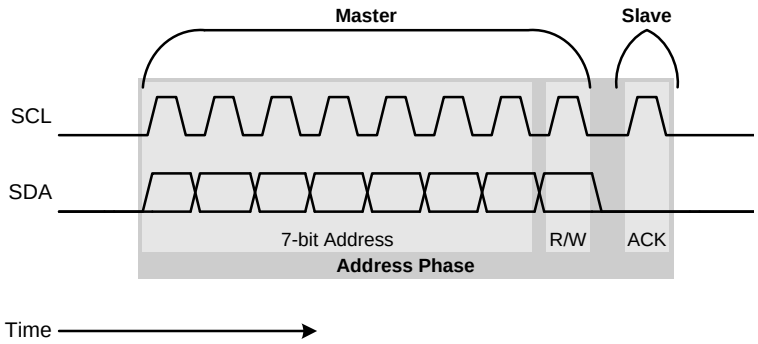


Figure 25.4. I2C 7-bit Address Phase

The 10-bit address phase consists of two bytes and two ACK or NACK responses from the slave. The first address byte contains bits 9 and 8 of the address (fixed value of 1111 0XX, where XX are the two address bits) and the R/W bit, which is the last bit of the byte. The rest of the address (bits 7:0) is transmitted in the second byte. All slaves that match the first part of the address will ACK after the first address byte. The one slave that also matches the second address byte will ACK after the second byte.

The address phase of a write operation using a 10-bit address consists of a single start followed by the two address bytes with R/W set to 0 for a write. The address phase of a read operation using a 10-bit address consists of a start followed by the two address bytes with R/W set to 0 for a write, followed by a repeated start with the first address byte and R/W set to 1 for a read, and the data for the master.

Figure 25.5 shows the 10-bit address phases for both a write and a read operation.

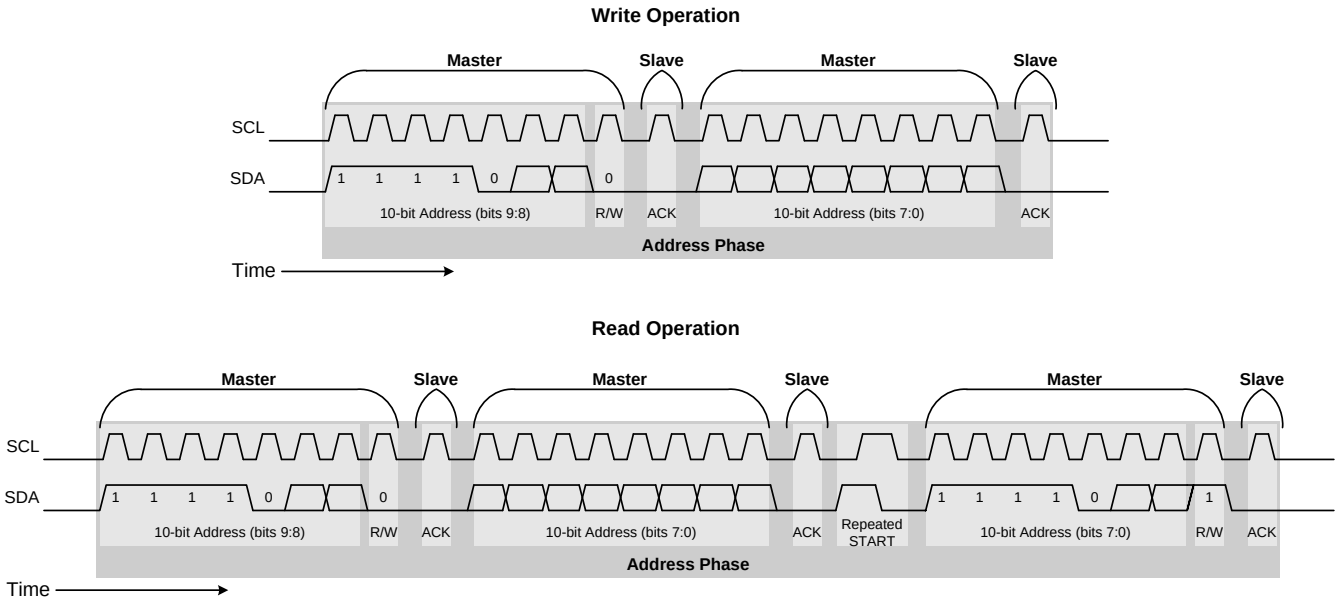


Figure 25.5. I2C 10-bit Address Phase

25.2.3. Data Phase

The data phase follows the address phase and is the same for both 7- and 10-bit address modes. The data phase consists of a byte of data followed back an ACK or NACK. In the case of a read operation, the slave provides the data byte and the master responds as shown in Figure 25.6. In the case of a write operation, the master provides the data byte and the slave responds as shown in Figure 25.7.

The entire data phase for a transaction can consist of one or more bytes. The master determines the total number of bytes sent by deciding when to send the stop condition that ends the transaction.

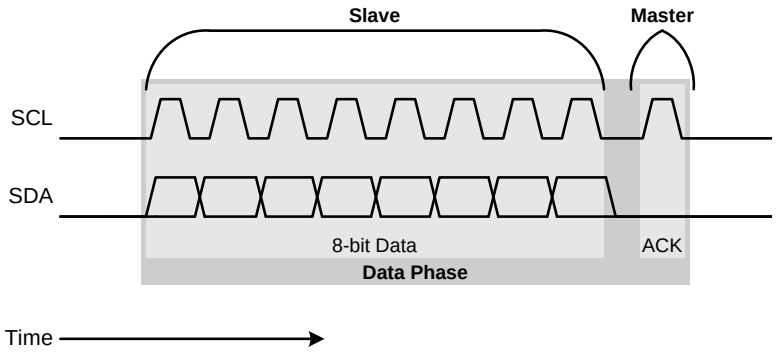


Figure 25.6. Single Byte I2C Read Data Phase

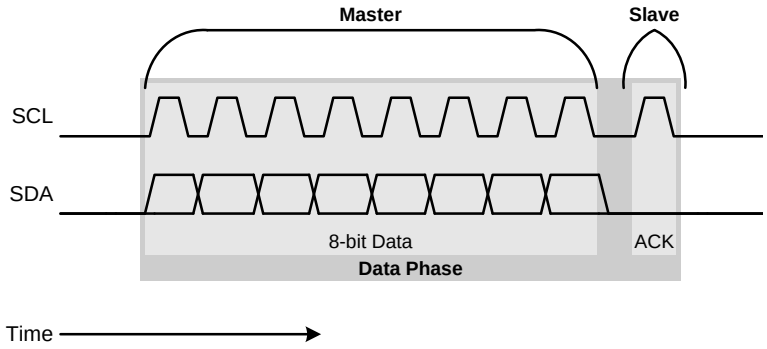


Figure 25.7. Single Byte I2C Write Data Phase

25.3. Clocking

The I2C module has an internal 6-bit APB clock divider that allows the module to support various APB and I2C clock frequencies. The divided I2C clock serves as the timebase for the SCL signal, rise and fall timing, and hardware-supported timeouts. The SCALER field in the CONFIG register sets the I2C module clock frequency (F_{I2C}) according to Equation 25.1.

$$F_{I2C} = \frac{F_{APB}}{(64 - SCALER)}$$

Equation 25.1. I2C Module Clock Frequency

25.3.1. Clock Setup

When operating as an I2C master, the I2C clock high and low times (T_{SCL_HIGH} and T_{SCL_LOW}) are based off the I2C module clock, and can be independently configured. The clock high time is determined by the T1RL field in the TIMERRL register, as shown in Equation 25.2. Likewise, the clock low time is determined by the SCLL field in the SCONFIG register, shown in Equation 25.3.

$$T_{SCL\_HIGH} = \frac{256 - T1RL}{F_{I2C}}$$

Equation 25.2. SCL High Time

$$T_{SCL\_LOW} = \frac{256 - SCLL}{F_{I2C}}$$

Equation 25.3. SCL Low Time

The I2C bus master controls the clock. However, slave devices have the option to extend the clock low time if needed. When operating as an I2C slave, the SCLL field defines a period for the device to hold SCL low after detecting a falling edge on the bus.

25.4. Operational Modes

The I2C module supports both master and slave states. Each step of the transaction process is controlled both by hardware and firmware to provide flexibility. A typical I2C module interrupt service routine contains several states to determine the module's next actions in response to activity on the bus. Each state must perform actions in the following order:

1. Set and clear bits for the next state.
2. Write or read from the DATA register.
3. Arm the transmit or receive operation.
4. Clear the pending interrupt flag.

The I2C module can send or receive multiple bytes autonomously during the data phase by setting the BC field in the CONFIG register to a value other than 1. If the BC value is set to a value other than 1, the transmit or receive interrupt will occur when all bytes have been sent or received (when BP is equal to BC). If multiple bytes are sent during a transaction using a BC value other than 1, the least-significant byte in the DATA register will be sent or received first.

The I2C module supports 7-bit addresses in hardware and can support 10-bit addresses by using firmware to transmit and decode the second address byte.

The basic software-controlled master and slave transactions are described in detail in the follow section. The additional features of the module are described in Section 25.6 and DMA modes are described in Section 25.8.

25.4.1. Master Write Transaction

In a master write transaction, the I2C module is in master mode and sends one or more bytes of data to an addressed slave on the bus.

The master write operation starts with firmware setting the STA bit to generate a start condition. A start interrupt occurs after the hardware transmits a start condition. The ISR or firmware routine should then clear the start bit (STA), set the targeted slave address and the R/W direction bit in the DATA register, set the byte count, arm the transmission (TXARM = 1), and clear the start interrupt.

After the hardware transmits the address and direction bit, a transmit interrupt occurs. The ISR should check the ACK bit to determine if the addressed slave acknowledged the address and is ready to receive data. If the master received a NACK, no slaves acknowledged the address, and firmware should set the STO bit to generate a stop. If the master received an ACK, the firmware should load the data into the DATA register, arm the transmission (TXARM = 1), and clear the transmit interrupt. This process is identical for each byte or set of bytes in the transmission.

For each set of bytes (determined by the BC field), the hardware generates an acknowledge interrupt if the slave NACKs a transmission, and will not generate an interrupt if the slave ACKs. If the slave NACKs, the master can choose to re-transmit the data or start the process over. To stop the current transmission, firmware should set the STO bit to generate a stop condition. When the stop interrupt occurs, the transmission can be re-started by setting the STA bit to generate a start condition.

When the hardware transmits the last data byte, a transmit interrupt occurs. The firmware can check the ACK or NACK status and should set the STO bit to generate a stop condition and clear the transmit interrupt before exiting the ISR. The transmission does not need to be armed in this case because an address or data isn't being sent.

A stop interrupt occurs after the hardware transmits the stop condition. Firmware should clear the stop interrupt and exit the ISR, completing the transaction.

Figure 25.8 shows a flow diagram of this master write transaction process.

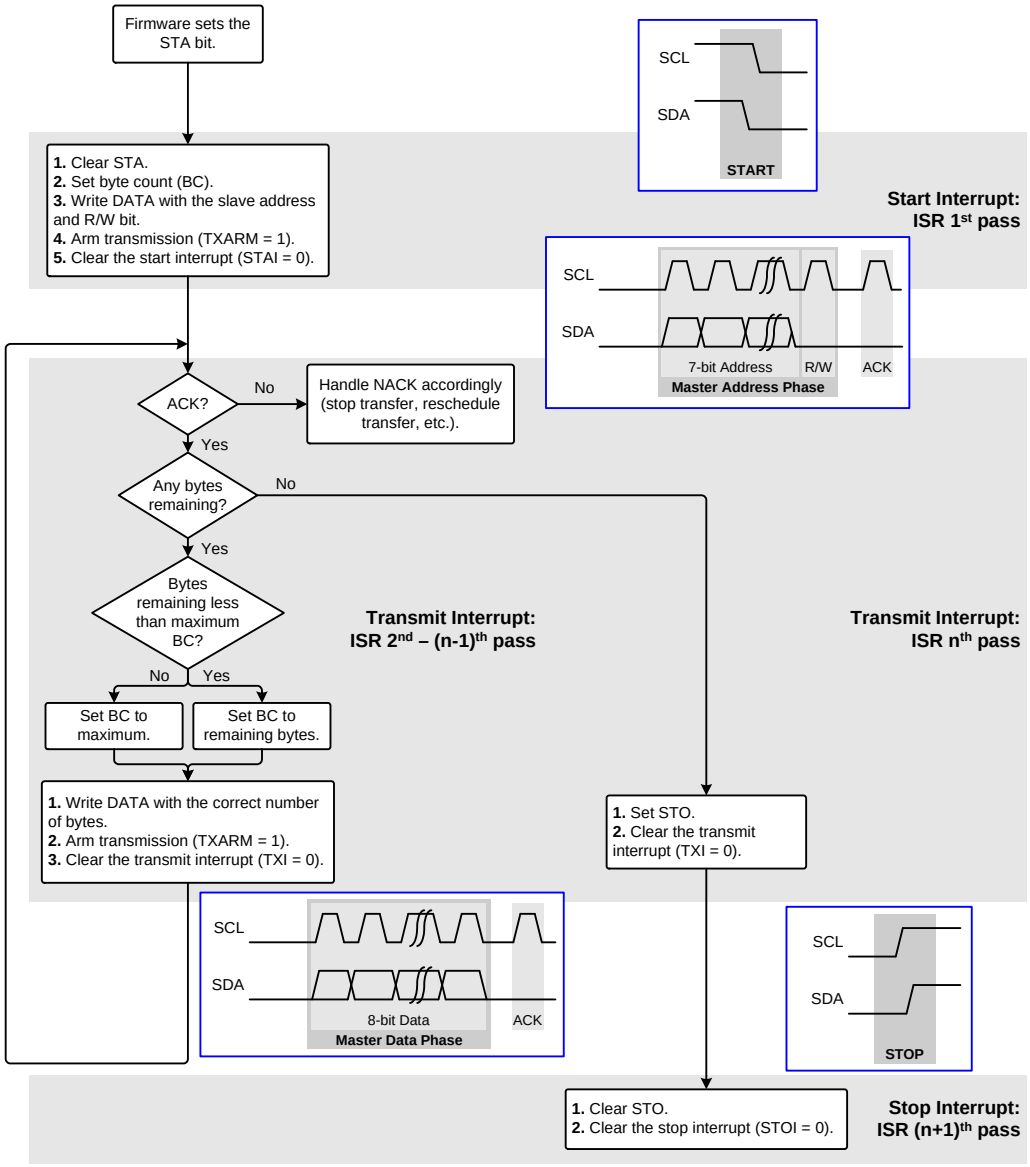


Figure 25.8. Master Write Flow Diagram (7-bit Address)

25.4.2. Master Read Transaction

In a master read transaction, the I2C module is in master mode and receives one or more bytes of data from an addressed slave on the bus.

The master read operation starts the same as the master write operation with firmware setting the STA bit to generate a start condition. A start interrupt occurs after the hardware transmits a start condition. The ISR or firmware routine should then clear the start bit (STA), set the targeted slave address and the R/W direction bit in the DATA register, set the byte count, arm the transmission (TXARM = 1), and clear the start interrupt.

After the hardware transmits the address and direction bit, a transmit interrupt occurs. The ISR should check the ACK bit to determine if the addressed slave acknowledged the address and is ready to receive data. If the master received a NACK, no slaves acknowledged the address, and firmware should set the STO bit to generate a stop. If the master received an ACK, the firmware should arm reception (RXARM = 1), set the byte count, and clear the transmit interrupt.

A master acknowledge interrupt occurs for each byte received from the slave (determined by the BC field). The firmware must set the ACK bit and clear the acknowledge interrupt.

When the master receives all of the bytes of the transfer (BP is equal to BC), a receive interrupt occurs. The firmware must read the data from the FIFO using the DATA register, set the BC field for the next reception, arm reception (RXARM = 1), and clear the receive interrupt. If the master does not need to read any additional data from the slave, the firmware should set the STO bit to generate a stop and clear the receive interrupt.

A stop interrupt occurs after the hardware transmits the stop condition. Firmware should clear the stop interrupt and exit the ISR, completing the transaction.

Figure 25.9 shows a flow diagram of this master read transaction process.

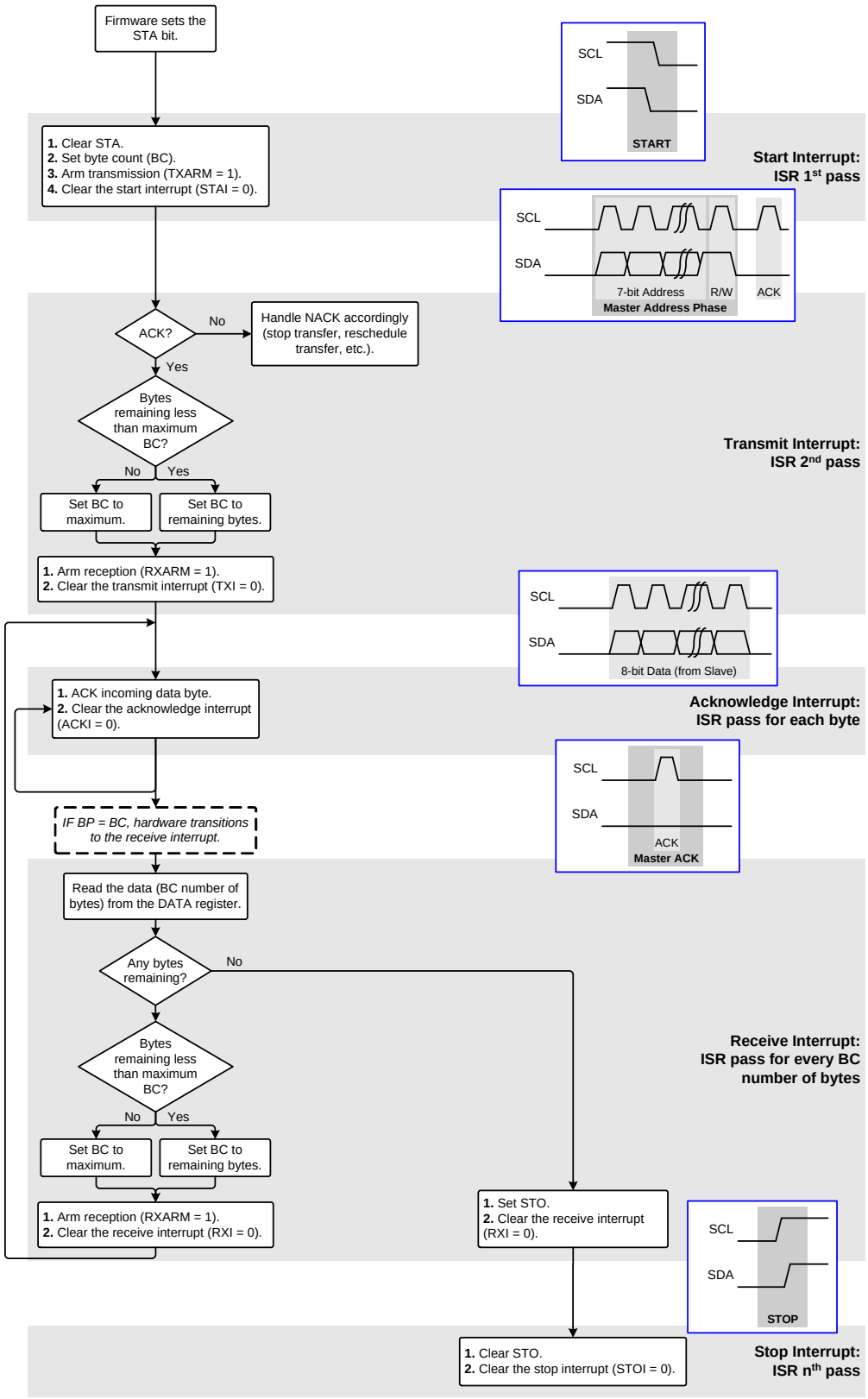


Figure 25.9. Master Read Flow Diagram (7-bit Address)

25.4.3. Repeated Starts

Repeated starts are used in master transactions when slaves require a write before the master can read data. The repeated start allows the master to maintain control of the bus even though two separate transactions take place.

The repeated start is the same as a master write transaction followed by a master read transaction where the write ends with another start instead of a stop condition.

The repeated start operation begins with firmware setting the STA bit to generate a start condition. A start interrupt occurs after the hardware transmits a start condition. The ISR or firmware routine should then clear the start bit (STA), set the targeted slave address and the R/W direction bit (write) in the DATA register, set the byte count, arm the transmission (TXARM = 1), and clear the start interrupt.

After the hardware transmits the address and direction bit, a transmit interrupt occurs. The ISR should check the ACK bit to determine if the addressed slave acknowledged the address and is ready to receive data. If the master received a NACK, no slaves acknowledged the address, and firmware should set the STO bit to generate a stop. If the master received an ACK, the firmware should write the data to the DATA register, set the byte count, arm the transmission (TXARM = 1), and clear the transmit interrupt.

The second transmit interrupt occurs after the master sends the data to the slave. The firmware should set the STA bit again and clear the transmit interrupt to generate the repeated start.

In the second start interrupt pass, the firmware should again write the slave address and R/W direction bit (read) in the DATA register before clearing the start bit, arming the transmission (TXARM = 1), setting the byte count, and clearing the start interrupt.

A master acknowledge interrupt occurs for each byte received from the slave (determined by the BC field). The firmware must set the ACK bit and clear the acknowledge interrupt.

When the master receives all of the bytes of the transfer (BP is equal to BC), a receive interrupt occurs. The firmware must read the data from the DATA register, set the BC field for the next reception, arm reception (RXARM = 1), and clear the receive interrupt. If the master does not need to read any additional data from the slave, the firmware should set the STO bit to generate a stop and clear the receive interrupt.

A stop interrupt occurs after the hardware transmits the stop condition. Firmware should clear the stop interrupt and exit the ISR, completing the transaction.

Figure 25.10 and Figure 25.11 show a flow diagram of the master repeated start process.

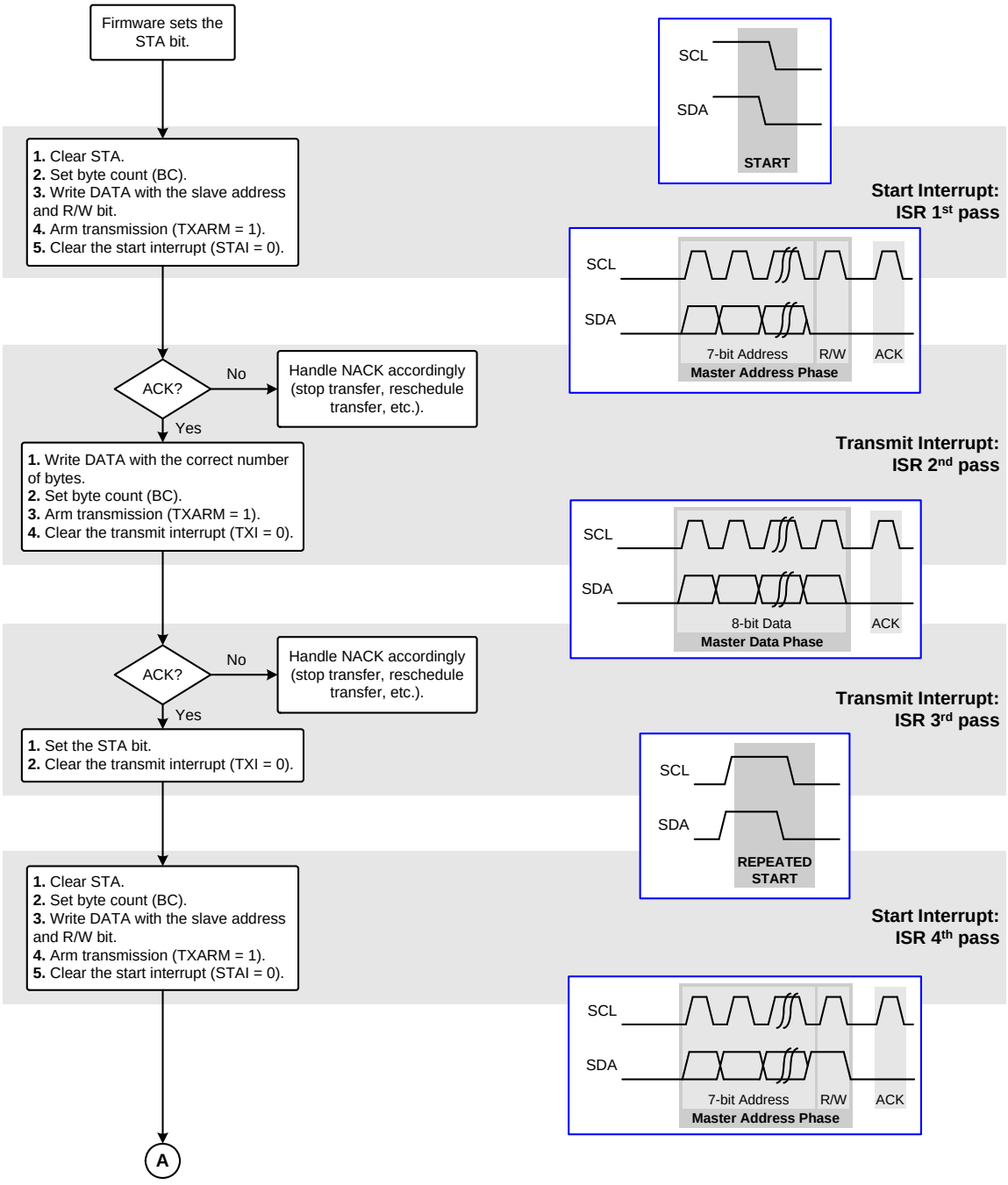


Figure 25.10. Master Repeated Start (Write/Read) Flow Diagram (7-bit Address) (Page 1/2)

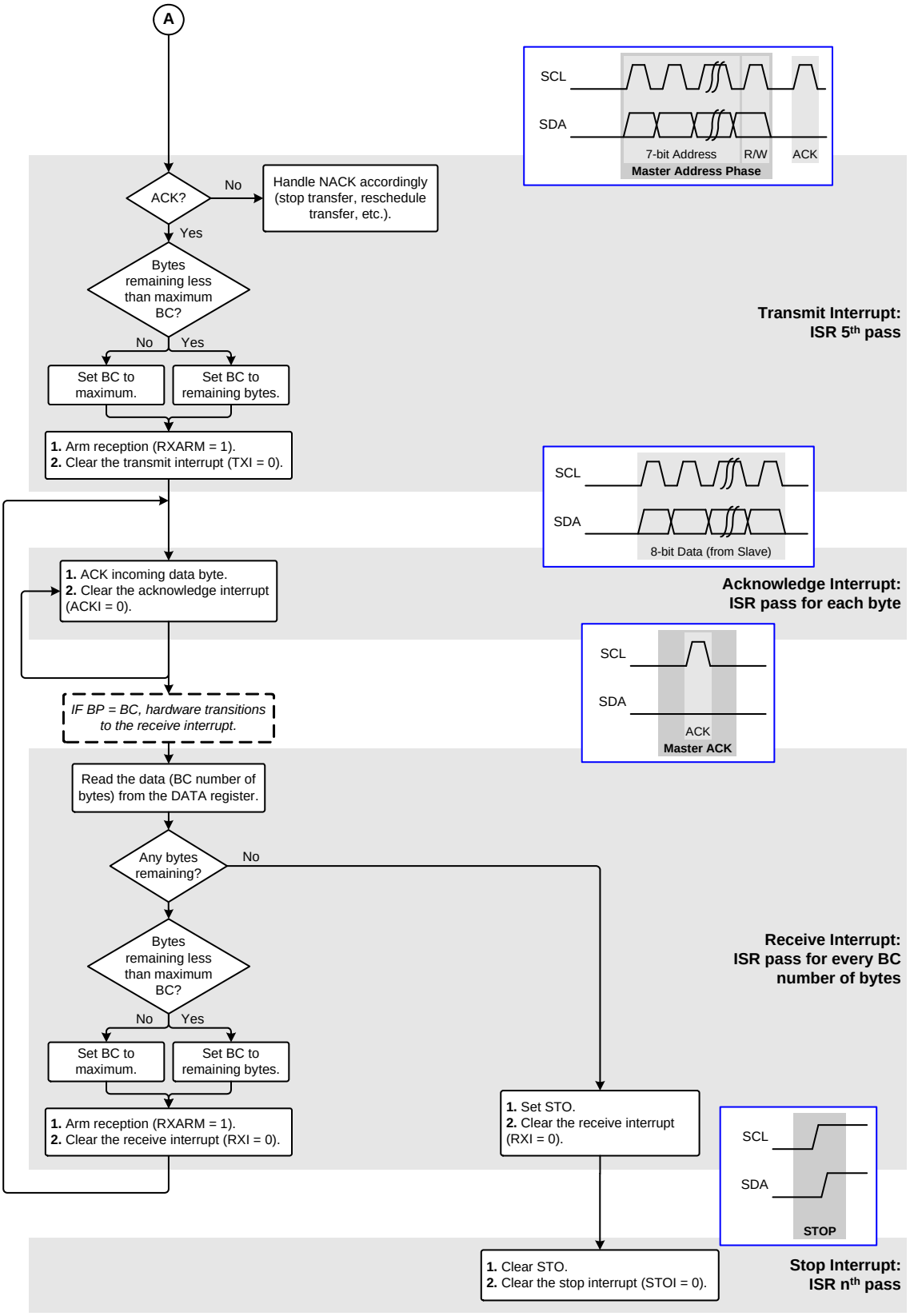


Figure 25.11. Master Repeated Start (Write/Read) Flow Diagram (7-bit Address) (Page 2/2)

25.4.4. Slave Write Transaction

In a slave write transaction, the I2C module is in slave mode and receives one or more bytes of data from a master.

The slave write operation starts with a start interrupt after a master generates a start and sends the address on the bus. The firmware should read the address and R/W direction bit from the DATA register and determine if the address matches the slave address. If not, the slave should send a NACK by clearing the ACK bit to 0 and clear the start interrupt. If the slave address does match, the slave should ACK the address by setting ACK to 1, set the byte count, clear the start bit, arm reception (RXARM = 1), and clear the start interrupt.

An acknowledge interrupt occurs each time the slave receives a byte. The firmware should respond with either an ACK or NACK and clear the acknowledge interrupt.

When the slave receives the last byte of the current packet (BP = BC), a receive interrupt occurs. The slave should read the data from the DATA register, set the byte count for the next set of bytes, arm reception, and clear the receive interrupt. If this is the last set of bytes the slave will receive, the slave should clear the receive interrupt.

A stop interrupt occurs after the master generates a stop condition on the bus. The firmware should clear the stop bit and the stop interrupt.

Figure 25.12 shows a flow diagram of this slave write transaction process.

25.4.5. Slave Read Transaction

In a slave read transaction, the I2C module is in slave mode and sends one or more bytes of data to a master.

The slave read operation starts with a start interrupt after a master generates a start and sends the address on the bus. The firmware should read the address and R/W direction bit from the DATA register and determine if the address matches the slave address. If not, the slave should send a NACK by clearing the ACK bit to 0 and clear the start interrupt. If the slave address does match, the slave should ACK the address by setting ACK to 1, set the byte count, write the data to the DATA register, clear the start bit, arm transmission (TXARM = 1), and clear the start interrupt.

A transmit interrupt occurs each time the slave sends a set of bytes (BP = BC). The firmware should check if there are any bytes remaining for the current transfer. If so, the byte count should be set, the new data written to the DATA register, transmission armed (TXARM = 1), and the transmit interrupt cleared. If no data remains to be sent, the slave should clear the transmit interrupt and wait for the master to generate a stop.

A stop interrupt occurs after the master generates a stop condition on the bus. The firmware should clear the stop bit and clear the stop interrupt.

Figure 25.13 shows a flow diagram of this slave read transaction process.

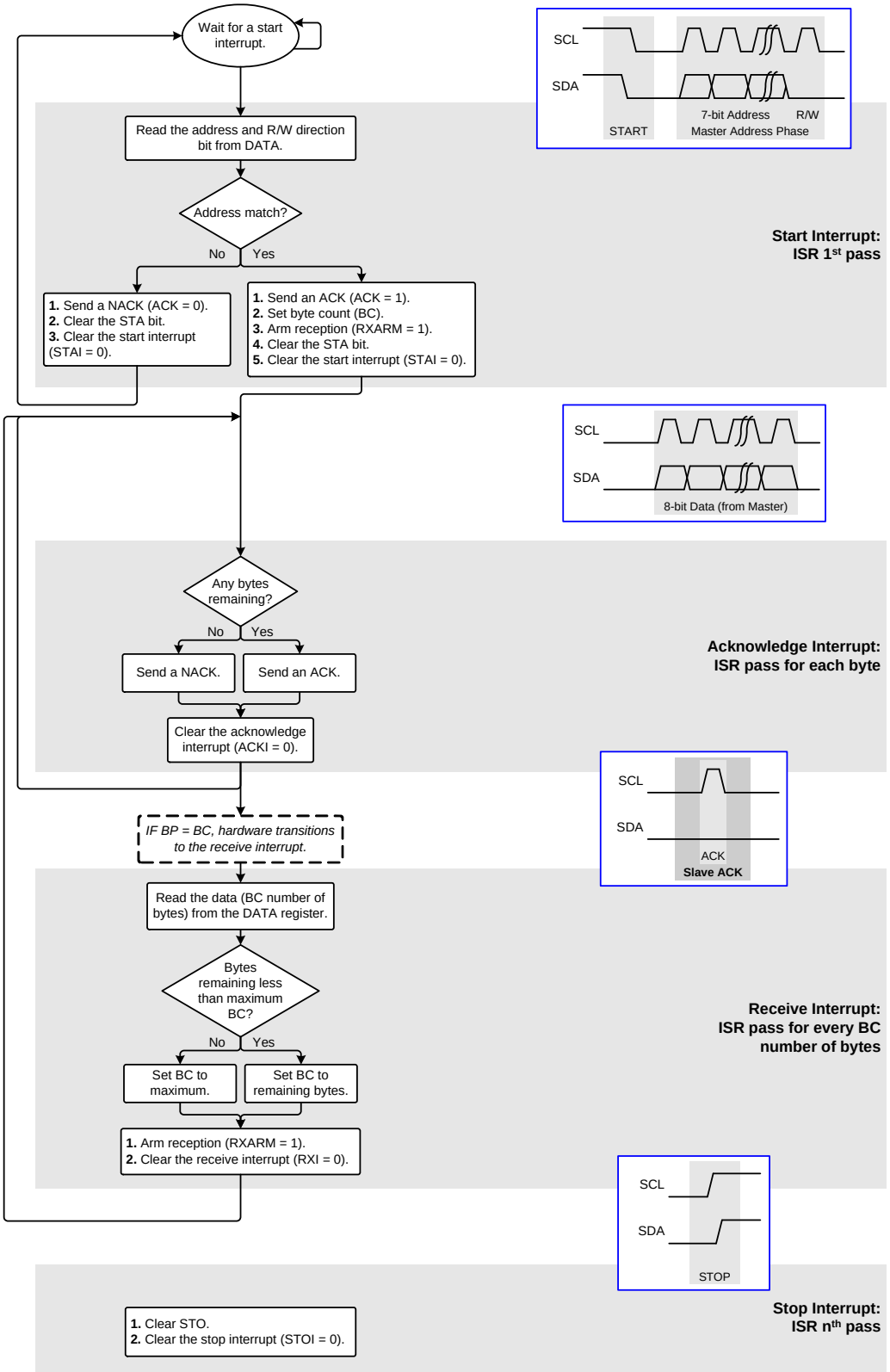


Figure 25.12. Slave Write Flow Diagram (7-bit Address)

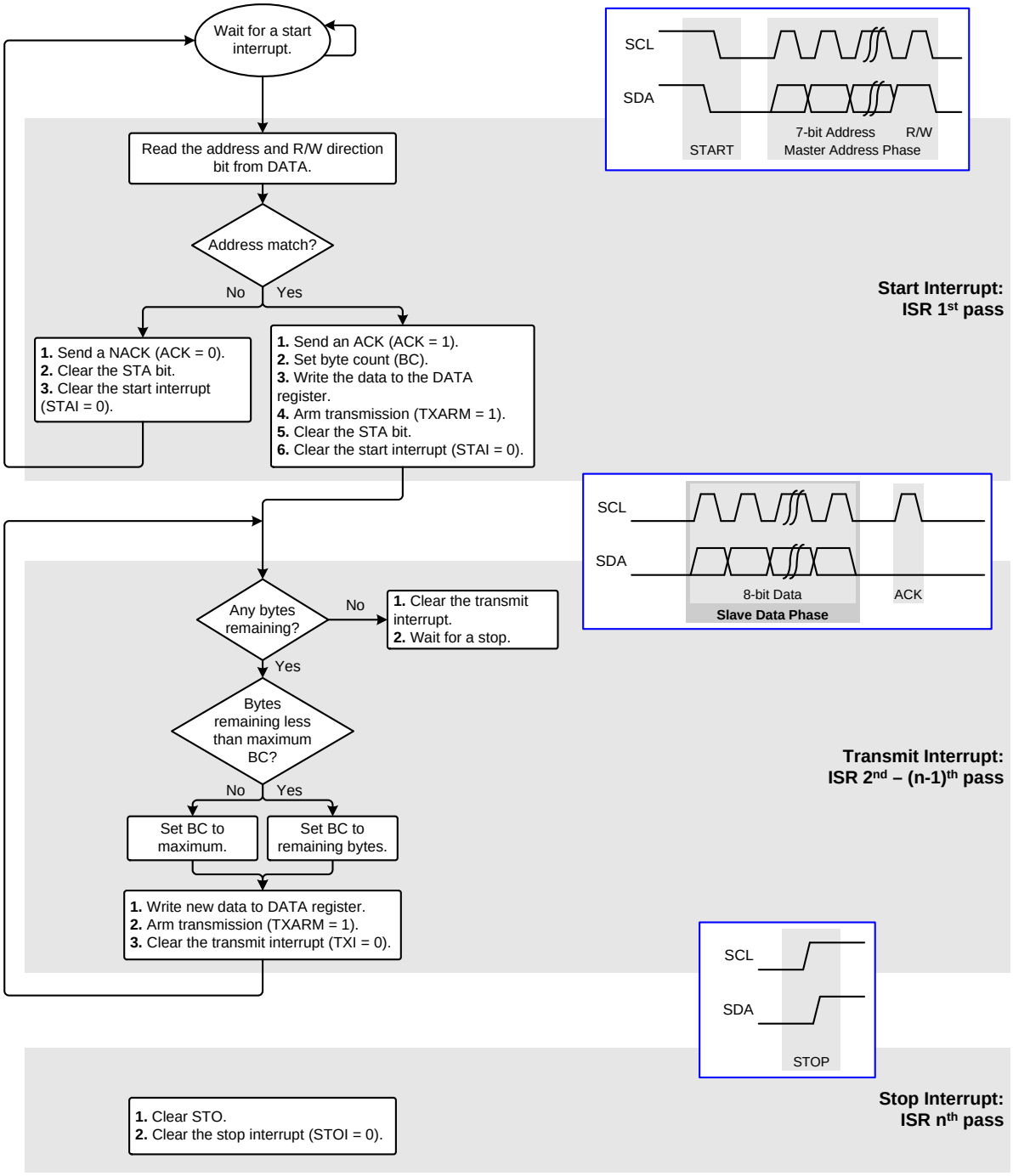


Figure 25.13. Slave Read Flow Diagram (7-bit Address)

25.5. Error Handling

25.5.1. Arbitration

A master may start a transfer only if the bus is free. The bus is free after a stop condition or after the SCL and SDA lines remain high for a specified time. In the event that two or more devices attempt to begin a transfer at the same time, the I2C protocol employs an arbitration scheme to force one master to give up the bus. The master devices continue transmitting until one attempts to transmit a 1 while the other attempts to transmit a 0. Since the bus is open-drain, the bus will be pulled low, and the master attempting to transmit the 1 will detect a low SDA signal and lose the arbitration. The winning master continues its transmission without interruption; the losing master becomes a slave (if slave states are supported) and receives the rest of the transfer, if addressed. This arbitration scheme is non-destructive: one device always wins, and no data is lost.

In the case of an arbitration lost event, the arbitration lost interrupt occurs. The read-only ARBLF flag mirrors the state of the ARBLI interrupt flag. This interrupt can occur in both master or slave mode whenever the I2C module is attempting to transmit data on the bus. When configured as a slave, the ARBLI interrupt flag indicates a protocol violation on the bus. In master mode, the firmware may want to save the aborted slave target and R/W direction bit to allow this attempt to be rescheduled after any pending slave response, if supported.

25.5.2. SMBus SCL Low Timeout

If the SCL line is held low by a device on the bus, no further communication is possible. Furthermore, the master cannot force the SCL line high to correct the error condition. To solve this problem, the SMBus protocol specifies that devices participating in a transfer must detect any clock cycle held low longer than 25 ms as a “timeout” condition. Devices that have detected the timeout condition must reset their communication interfaces no later than 10 ms after detecting the timeout condition.

When the I2C module is enabled (I2CEN = 1), the dedicated I2C 32-bit timer provides a counter for SCL low timeout detection. The SCL low timeout counter is 20 bits, with Timer Byte [3: 2] providing the upper 16 bits of the counter. The least significant bits of the timeout counter are not available to program. The timeout counter can be set by setting the T3:T2 and T3RL:T2RL fields. The hardware automatically forces the timer to reload when SCL is high and allows the timer to count when SCL is low.

In the event that T3:T2 matches the value in T3RL:T2RL, the timeout T3I flag will be set and an interrupt will be generated, if enabled. If this interrupt occurs, firmware can reset the I2C module using the RESET bit.

25.5.3. Bus Free Timeout

The I2C bus is free if all previous transactions ended with a stop condition and both SCL and SDA are high. The I2C module supports a bus free timeout that detects if SCL and SDA are high for the specified timeout period without a stop condition appearing on the bus.

The dedicated I2C Timer Byte 0 serves as the bus free timeout counter when the module is enabled (I2CEN = 1). The T0 field contains the timeout value and T0RL contains the counter reload value. T0 will count up if both SCL and SDA are high, and hardware forces T0 to reload from T0RL if SCL or SDA are low. If the I2C module is waiting for the bus to be free before starting a transfer, the module will generate the start condition after the timeout period expires.

25.6. Additional Features

25.6.1. Hardware Acknowledge and General Call Addressing

When the HACKEN bit in the CONTROL register is cleared to 0, the firmware on the device must detect incoming slave addresses and ACK or NACK the slave address and incoming data bytes. As a receiver, writing the ACK bit defines the outgoing ACK value; as a transmitter, reading the ACK bit indicates the value received during the last ACK cycle. ACKRQF is set each time a byte is received, indicating that an outgoing ACK value is needed. When ACKRQF is set, firmware should write the desired outgoing value to the ACK bit before clearing the appropriate interrupt flag. A NACK will be generated if software does not write the ACK bit before clearing the interrupt. SDA will reflect the defined ACK value immediately following a write to the ACK bit; however, SCL will remain low until firmware clears the interrupt. If a received slave address is not acknowledged, further slave events will be ignored until the next start is detected.

The hardware acknowledge feature (enabled when HACKEN = 1) provides for automatic 7-bit slave address recognition and ACK generation. The ADDRESS and MASK fields define which addresses are automatically recognized by the hardware. A single address or range of addresses (including the General Call Address of all 0's) can be specified using these two registers. A 1 in a bit position of the slave address mask (MASK) enables a comparison between the received slave address and the hardware's slave address (ADDRESS) for those bits. A 0 in the slave address mask means that bit will be treated as a "don't care" for comparison purposes. Additionally, hardware will recognize the General Call Address (all 0's) if the GCEN bit in the CONTROL register is set to 1. Firmware can use the SLVAF bit to determine if the slave recognized the slave address or the General Call. Table 25.1 shows some example parameter settings and the slave addresses that will be recognized by hardware.

Table 25.1. Hardware Address Recognition Examples (HACKEN = 1)

Hardware Slave Address (ADDRESS)	Hardware Slave Address Mask (MASK)	General Call Address (GCEN)	Slave Addresses Recognized by Hardware
0x34	0x7F	Disabled	0x34
0x34	0x7F	Enabled	0x34, 0x00
0x34	0x7E	Disabled	0x34, 0x35
0x34	0x7E	Enabled	0x34, 0x35, 0x00
0x70	0x73	Disabled	0x70, 0x74, 0x78, 0x7C

During the data phases of the transfer, the I2C hardware in receiver mode will use the value currently specified by the ACK bit to automatically respond during the ACK cycle of an incoming data byte. As a transmitter, reading the ACK bit indicates the value received on the last ACK cycle. The ACKRQF bit is not used when hardware ACK generation is enabled. If a received slave address is NACKed by hardware, the hardware will ignore any further slave events until the next start is detected and will not generate a start interrupt.

When hardware acknowledge is enabled in receive mode, the last byte acknowledge enable bit (LBACKEN) can be used to automatically NACK the last byte of a transfer, if desired.

25.6.1.1. Automatic Transmit or Receive Enable

The automatic transmit or receive mode can only be used if hardware acknowledge is enabled. If the ATXRXEN bit is set to 1, the I2C module will automatically ACK the incoming address and switch to either receive or transmit mode depending on the R/W bit. Enabling automatic transmit or receive mode bypasses the start interrupt, so the receive or transmit interrupt is the first interrupt triggered if ATXRXEN is set to 1.

25.6.2. SDA Setup and Hold Time Extensions

The data setup and hold times can be optionally extended using the SETUP and HOLD fields in the SCONFIG register. I2C Timer Byte 0 determines the data setup and hold times if SETUP and HOLD are 0. If SETUP or HOLD is set to a non-zero value, this setting overrides the I2C Timer Byte 0 count.

These extensions are based on the I2C module clock, and the equations for the additional time are provided in the register descriptions for these fields.

25.6.3. General Purpose Timer

When the I2C is not in use (I2CEN = 0), the dedicated I2C 32-bit timer that generates SCL and SDA timing can be used as an additional general purpose count-up timer.

This I2C timer can be configured to the following modes:

- Mode 0: One 32-bit timer with auto-reload (Timer Bytes [3: 0]).
- Mode 1: Two 16-bit timers with auto-reload (high timer: Timer Bytes [3: 2], low timer: Timer Bytes [1: 0]).
- Mode 2: Four 8-bit timers with auto-reload (Timer Byte 3, Timer Byte 2, Timer Byte 1, and Timer Byte 0).
- Mode 3: One 16-bit and two 8-bit timers with auto-reload (Timer Bytes [3: 2], Timer Byte 1, and Timer Byte 0).

The mode of the timers can be controlled using the TMD field. The TxRUN bits control the timers, TxIEN bits enable the timer interrupts, Tx fields contain the current timer value, and TxRL fields contain the reload values. When one of the timers overflows (from all 1's to all 0's), the appropriate TxI interrupt flag will be set and an I2C interrupt will occur, if enabled. The timer run bits (TxRUN) are gated by a global I2C timer enable bit (TIMEREN) that must be set to 1 for the timers to count on the I2C module clock.

25.7. Debug Mode

Firmware can set the DBGMD bit to force the I2C module to halt on a debug breakpoint. The I2C block will complete the current byte transfer before halting. Clearing the DBGMD bit forces the module to continue operating while the core halts in debug mode.

25.8. DMA Configuration and Usage

The DMA interface to the I2C block has one channel mapped to transmit and one channel mapped to receive operations. When DMA is enabled (DMAEN = 1), data must be transferred 4 bytes at a time. For a transfer length which is not a multiple of 4, the extra bytes in the MSB of the final DMA transfer are ignored.

The I2C module can be programmed to transfer up to 255 bytes in a single DMA operation by setting the DMALEN field in the I2CDMA register. For transmit operations, the DMA engine requests a word transfer to DATA whenever the DATA register is empty and the DMALEN field is not 0. For receive operations, the DMA engine requests a word transfer from DATA whenever the DATA register is full and the DMALEN field is not 0. The DMALEN field will decrement by 1 when the hardware transmits or receives a byte. If the DMALEN value is less than 4, the DMA transmits or receives the remaining bytes only. A transmit or receive interrupt occurs when DMALEN is zero.

If the hardware receives a NACK during a DMA operation, it will generate an acknowledge interrupt and stop the remaining transaction. If the hardware detects a stop condition before DMALEN is zero, the transfer stops and a stop interrupt occurs.

The I2C Module DMA configuration is shown in Figure 25.14.

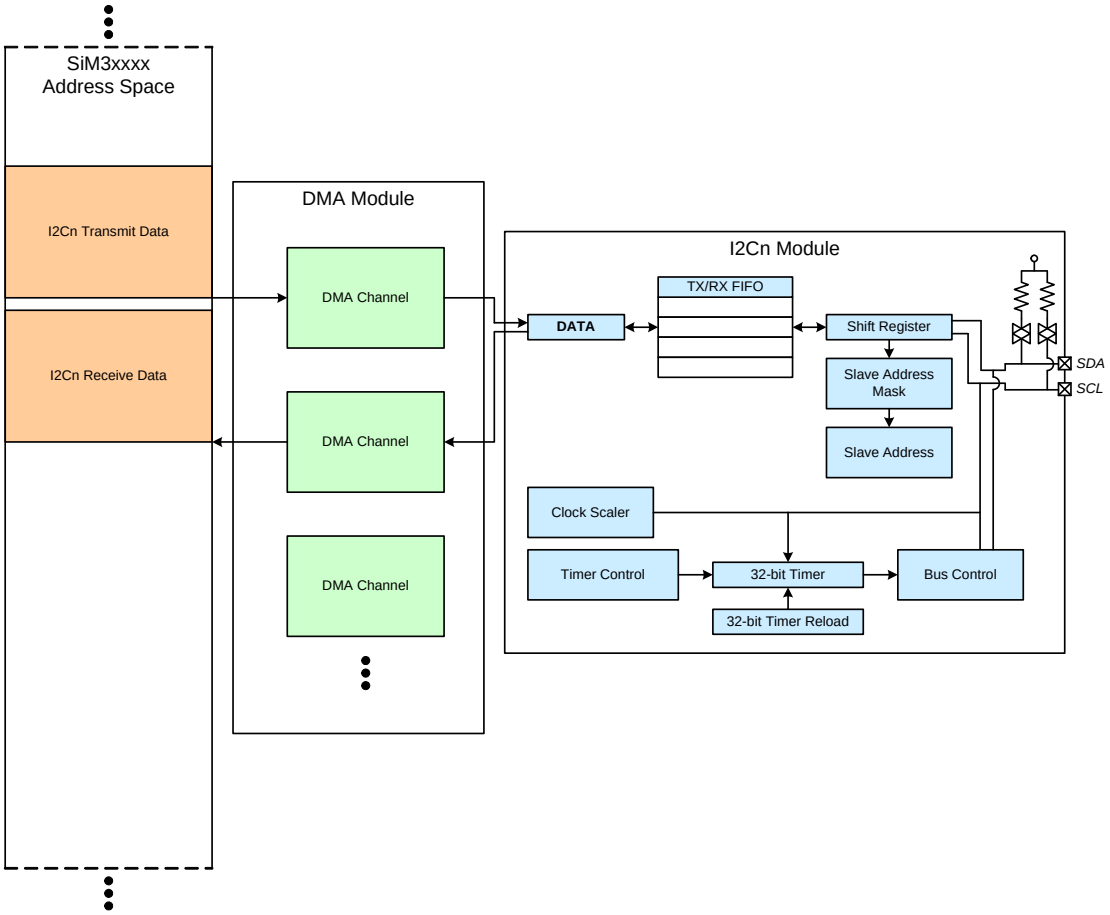


Figure 25.14. I2C Module DMA Configuration

25.8.1. Automatic Hardware DMA Options

To further automate the I2C transfer, firmware can enable Hardware Acknowledge (HACKEN = 1), Automatic Transmit or Receive Enable (ATXRXEN = 1), or Last Byte Acknowledge Enable (LBACKEN = 1) in DMA mode. If all of these modes are enabled, the hardware will automatically acknowledge any received bytes, automatically switch to transmit or receive mode depending on the set direction of the R/W bit, and ACK or NACK the last byte of the transfer, if it's a receive operation.

25.8.2. Master Write with DMA and Automatic Hardware Enabled

For a master write operation with all automatic modes enabled, the firmware should:

1. Set HACKEN and ATXRXEN to 1.
2. Write the address and R/W bit into DATA.
3. Program the DMALEN field to the appropriate value.
4. Set up the DMA transmit channel appropriately.
5. Enable the DMA mode in the I2C module (DMAEN = 1).
6. Issue a start by setting STA to 1.

If the master does not receive a NACK from the slave, the first interrupt received will be the transmit interrupt after all bytes are transferred. The master can continue to send more bytes by setting up another transfer or issue a stop to end the transfer. Once the stop interrupt occurs, the firmware must clear the STO bit to 0 and clear the stop interrupt.

The DMA master write operation is shown in Figure 25.15.

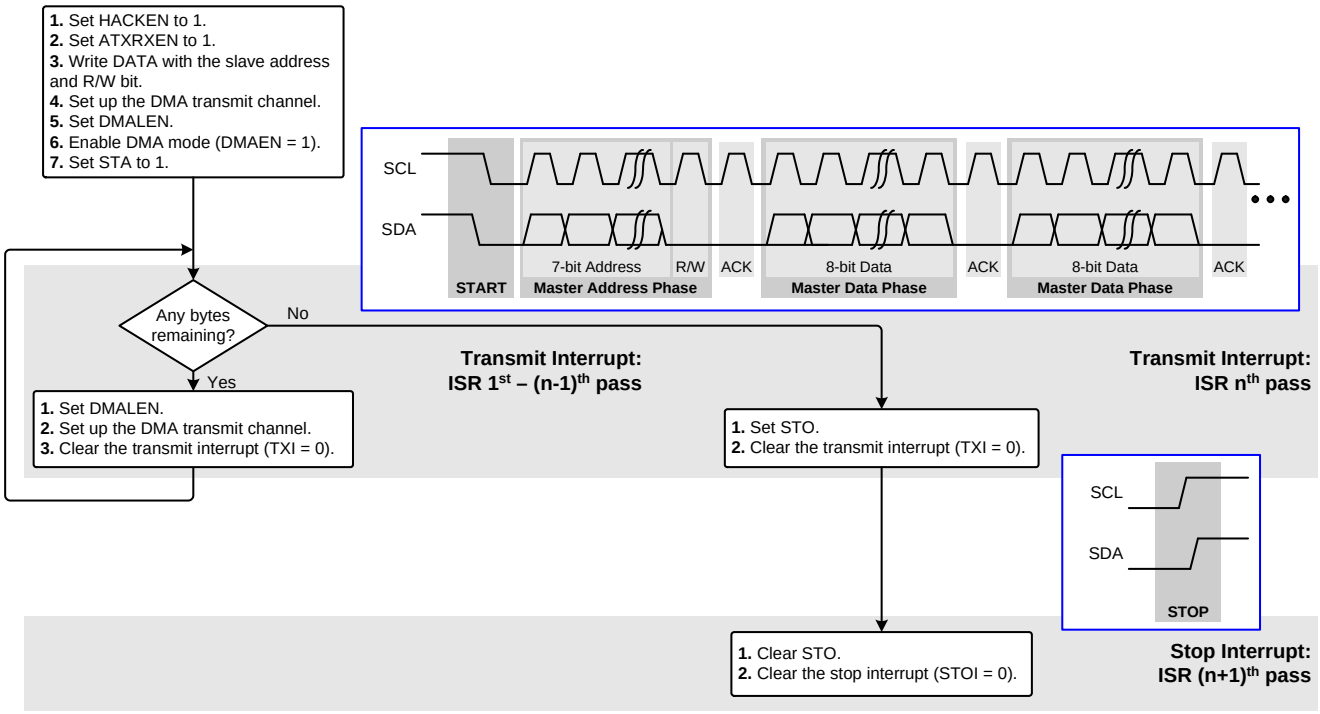


Figure 25.15. Master Write with DMA and Automatic Hardware Flow Diagram (7-bit Address)

25.8.3. Master Read with DMA and Automatic Hardware Enabled

For a master read operation with all automatic modes enabled, the firmware should:

1. Set HACKEN and ATXRXEN and to 1.
2. Write the address and R/W bit into DATA.
3. Program the DMALEN field to the appropriate value.
4. Set up the DMA receive channel appropriately.
5. Set LBACKEN to the appropriate value for the transfer.
6. Enable the DMA mode in the I2C module (DMAEN = 1).
7. Issue a start by setting STA to 1.

If the master does not receive a NACK from the slave, the first interrupt received will be the receive interrupt after all bytes are transferred. The master can continue to receive more bytes by setting up another transfer or issue a stop to end the transfer. Once the stop interrupt occurs, the firmware must clear the STO bit to 0 and clear the stop interrupt.

This DMA master read operation is shown in Figure 25.16.

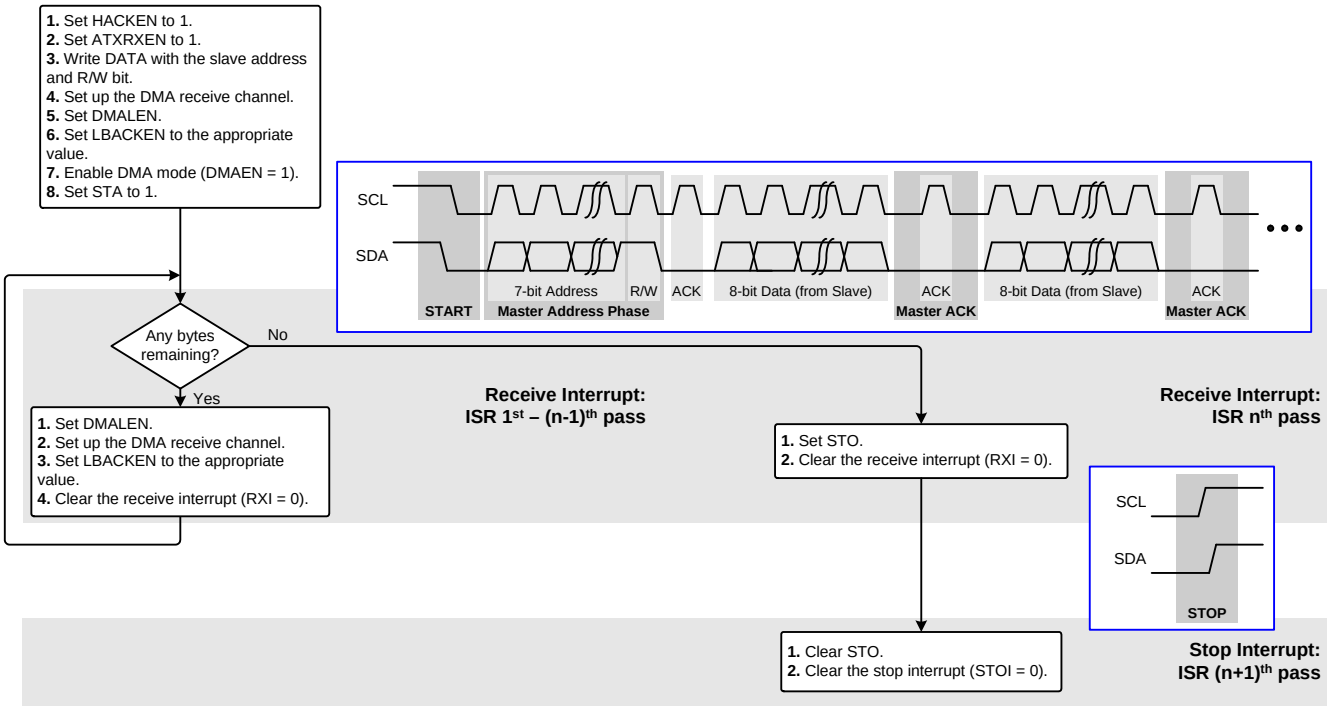


Figure 25.16. Master Read with DMA and Automatic Hardware Flow Diagram (7-bit Address)

25.8.4. Slave Write with DMA and Automatic Hardware Enabled

The DMA slave write firmware procedure with automatic hardware enabled is as follows:

1. Program the slave address and mask in ADDRESS and MASK.
2. Set the DMALEN field.
3. Set up the DMA receive channel appropriately.
4. Enable the DMA mode in the I2C module (DMAEN = 1).

The first interrupt the slave will receive is a receive interrupt, since all the bytes are automatically acknowledged. If more bytes should be received from the master, the firmware can set up another DMA transfer by resetting the DMA channel, reprogramming DMALEN and clearing the receive interrupt.

When the slave receives the stop interrupt, the firmware should clear the STO bit and the stop interrupt to end the transfer.

The DMA slave write operation is shown in Figure 25.17.

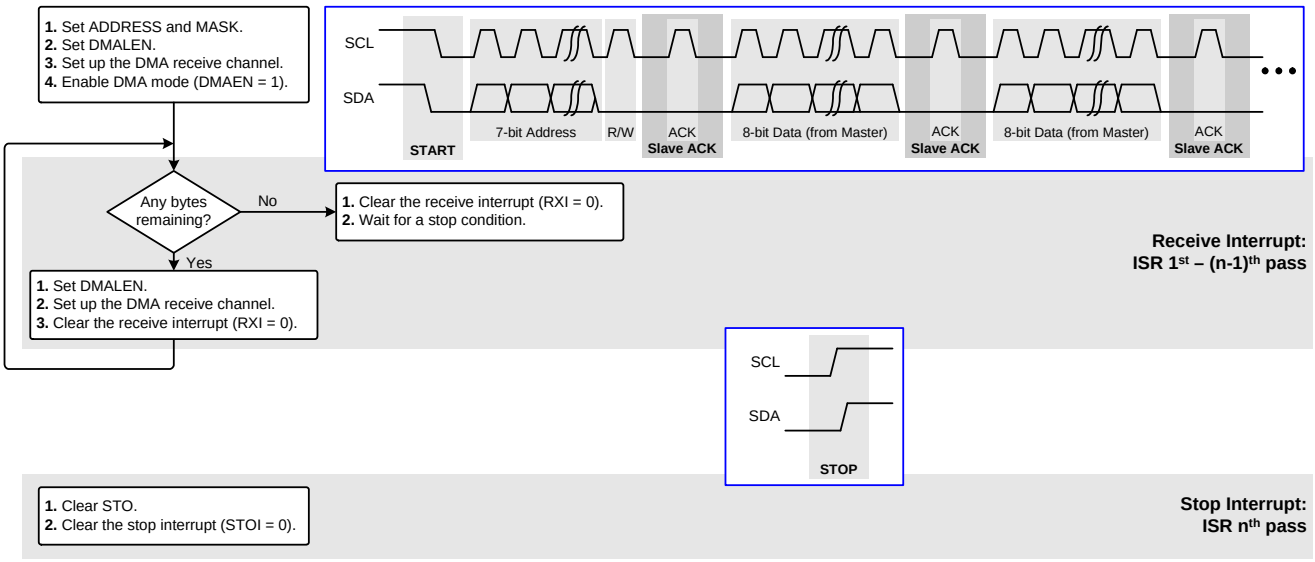


Figure 25.17. Slave Write with DMA and Automatic Hardware Flow Diagram (7-bit Address)

25.8.5. Slave Read with DMA and Automatic Hardware Enabled

The DMA slave read firmware procedure with automatic hardware enabled is as follows:

1. Program the slave address and mask in ADDRESS and MASK.
2. Set the DMALEN field.
3. Set up the DMA transmit channel appropriately.
4. Enable the DMA mode in the I2C module (DMAEN = 1).

The first interrupt the slave will receive is a transmit interrupt after all bytes are sent to the master. If more bytes should be sent to the master, the firmware can set up another DMA transfer by resetting the DMA channel, reprogramming DMALEN, and clearing the transmit interrupt.

If no more data is to be sent, when the slave receives the stop interrupt, the firmware should clear the STO bit and the stop interrupt to end the transfer.

The DMA slave read operation is shown in Figure 25.18.

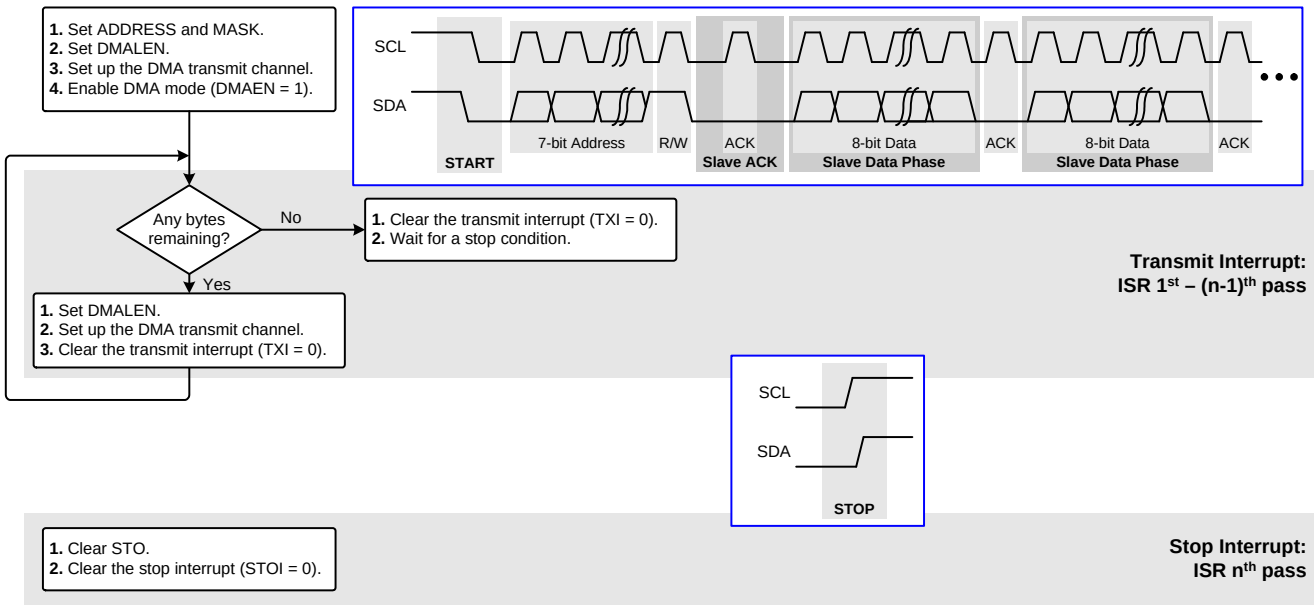


Figure 25.18. Slave Read with DMA and Automatic Hardware Flow Diagram (7-bit Address)

25.9. I2C0 Registers

This section contains the detailed register descriptions for I2C0 registers.

Register 25.1. I2C0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	I2CEN	RESET	GCEN	Reserved	LBACKEN	Reserved	HACKEN	SMINH	DBGMD	FMD	ATXRXEN	SLVAF	TXARM	RXARM	T3I	T2I
Type	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	R	RW	RW	RW	RW
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1I	T0I	ARBLI	STAI	TXI	RXI	ACKI	STOI	MSMDF	TXMDF	STA	STO	ACKRQF	ARBLF	ACK	BUSYF
Type	RW	RW	RW	RW	RW	RW	RW	RW	R	R	RW	RW	R	R	RW	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_CONTROL = 0x4000_9000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 25.2. I2C0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	I2CEN	**I2C Enable.** 0: Disable the I2C module. 1: Enable the I2C module.
30	RESET	**Module Soft Reset.** The following bits and fields are inaccessible while the module is in soft reset (RESET = 1): all interrupt flags (TXI, RXI, STAI, STOI, ACKI, ARBLI, T0I, T1I, T2I, T3I), STA, STO, TXARM, RXARM, ACK, ACKRQF, DMALEN, DATA, TIMER, and SCLLTIMER. 0: I2C module is not in soft reset. 1: I2C module is in soft reset and firmware cannot access all bits in the module.
29	GCEN	**General Call Address Enable.** 0: Disable General Call address decoding. 1: Enable General Call address decoding.
28	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 25.2. I2C0_CONTROL Register Bit Descriptions

Bit	Name	Function
27	LBACKEN	**Last Byte Acknowledge Enable.** Automatic hardware acknowledge mode must be enabled (HACKEN = 1) for this bit setting to have an effect. 0: NACK after the last byte is received. 1: ACK after the last byte is received.
26	Reserved	Must write reset value.
25	HACKEN	**Auto Acknowledge Enable .** 0: Disable automatic hardware acknowledge. 1: Enable automatic hardware acknowledge.
24	SMINH	**Slave Mode Inhibit.** 0: Enable Slave modes. 1: Inhibit Slave modes. The module will not respond to a Master on the bus.
23	DBGMD	**I2C Debug Mode.** 0: The I2C module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the I2C module to halt.
22	FMD	**Filter Mode.** 0: Disable the input filter. 1: Enable the input filter.
21	ATXRXEN	**Auto Transmit or Receive Enable.** 0: Do not automatically switch to transmit or receive mode after a Start. 1: If automatic hardware acknowledge mode is enabled (HACKEN = 1), automatically switch to transmit or receive mode after a Start.
20	SLVAF	**Slave Address Type Flag.** 0: Slave address detected. 1: General Call address detected.
19	TXARM	**Transmit Arm.** 0: Disable data and address transmission. 1: Enable the module to perform a transmit operation.
18	RXARM	**Receive Arm.** 0: Disable data and address reception. 1: Enable the module to perform a receive operation.
17	T3I	**I2C Timer Byte 3 Interrupt Flag.** When the I2C module is enabled (I2CEN = 1), this interrupt flag will be set to 1 if an SCL low timeout occurs. When using the I2C Timer as a stand-alone timer (when I2C is disabled) this interrupt flag will set if an overflow occurs out of the the I2C Timer Byte 3. Writing a 1 to this bit will manually trigger the interrupt. This flag must be cleared by firmware.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 25.2. I2C0_CONTROL Register Bit Descriptions

Bit	Name	Function
16	T2I	**I2C Timer Byte 2 Interrupt Flag.** When using the I2C Timer as a stand-alone timer (when I2C is disabled) this interrupt flag will set if an overflow occurs out of the the I2C Timer Byte 2. Writing a 1 to this bit will manually trigger the interrupt. This flag must be cleared by firmware.
15	T1I	**I2C Timer Byte 1 Interrupt Flag.** When using the I2C Timer as a stand-alone timer (when I2C is disabled) this interrupt flag will set if an overflow occurs out of the I2C Timer Byte 1. Writing a 1 to this bit will manually trigger the interrupt. This flag must be cleared by firmware.
14	T0I	**I2C Timer Byte 0 Interrupt Flag.** When using the I2C Timer as a stand-alone timer (when I2C is disabled) this interrupt flag will set if an overflow occurs out of the I2C Timer Byte 0. Writing a 1 to this bit will manually trigger the interrupt. This flag must be cleared by firmware.
13	ARBLI	**Arbitration Lost Interrupt Flag.** This bit is set to 1 by hardware when an arbitration lost condition occurs. This bit must be cleared by firmware.
12	STAI	**Start Interrupt Flag.** This bit is set to 1 by hardware when a start or repeated start condition occurs. The STO bit is also set with a repeated start to differentiate from a normal start condition. This bit must be cleared by firmware.
11	TXI	**Transmit Done Interrupt Flag.** This bit is set to 1 by hardware when a the module is transmitting data and BP is equal to BC. This bit must be cleared by firmware.
10	RXI	**Receive Done Interrupt Flag.** This bit is set to 1 by hardware when a the module is receiving data and BP is equal to BC. This bit must be cleared by firmware.
9	ACKI	**Acknowledge Interrupt Flag.** This bit is set to 1 by hardware when an acknowledge phase occurs and requires a response. This bit must be cleared by firmware.
8	STOI	**Stop Interrupt Flag.** This bit is set to 1 by hardware when a stop condition occurred or was generated. This bit must be cleared by firmware.
7	MSMDF	**Master/Slave Mode Flag.** 0: Module is operating in Slave mode. 1: Module is operating in Master mode.
6	TXMDF	**Transmit Mode Flag.** 0: Module is in receiver mode. 1: Module is in transmitter mode.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 25.2. I2C0_CONTROL Register Bit Descriptions

Bit	Name	Function
5	STA	**Start.** This bit indicates whether hardware generated or detects a start on the bus. This bit should be cleared to 0 by firmware after a start is detected. Setting this bit to 1 generates a start condition in master mode.
4	STO	**Stop.** This bit indicates whether hardware generated or detects a stop on the bus. This bit should be cleared to 0 by firmware after a stop is detected. Setting this bit to 1 generates a stop condition in master mode.
3	ACKRQF	**Acknowledge Request Flag.** 0: ACK has not been requested. 1: ACK requested.
2	ARBLF	**Arbitration Lost Flag.** This read-only flag mirrors the state of the ARBLI interrupt flag. 0: Arbitration lost error has not occurred. 1: Arbitration lost error occurred.
1	ACK	**Acknowledge.** Reading this bit returns the receive status of an ACK. Writing this bit to 1 sets the hardware to transmit an ACK.
0	BUSYF	**Busy Flag.** The BUSYF flag is set to 1 by hardware when a Start is generated or detected. This flag is cleared to 0 when hardware generates or detects a Stop or senses a bus-free timeout condition. 0: A transaction is not currently taking place. 1: A transaction is currently taking place.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 25.2. I2C0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TIMEREN	Reserved	TMD		T3RUN	T2RUN	T1RUN	T0RUN	BP		BC		Reserved		T3IEN	T2IEN
Type	RW	R	RW		RW	RW	RW	RW	R		RW		R		RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1IEN	T0IEN	ARBLIEN	STAIEN	TXIEN	RXIEN	ACKIEN	STOIEN	Reserved		SCALER					
Type	RW	RW	RW	RW	RW	RW	RW	RW	R		RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_CONFIG = 0x4000_9010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 25.3. I2C0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	TIMEREN	**I2C Timer Enable.** This bit can only be set when using the I2C Timer Bytes 0-3 for general purpose use. If I2CEN is set to 1, this bit should always be cleared to 0. If I2CEN is cleared to 0, this bit can be set to 1 to start the timer running. 0: Disable I2C Timer. 1: Enable I2C Timer for general purpose use. This setting should not be used when the I2C module is enabled (I2CEN = 1).
30	Reserved	Must write reset value.
29:28	TMD	**I2C Timer Mode.** This setting only takes effect when using the I2C timer as a stand-alone clock source and the I2C module is disabled (I2CEN = 0). 00: I2C Timer Mode 0: Operate the I2C timer as a single 32-bit timer : Timer Bytes [3 : 2 : 1 : 0]. 01: I2C Timer Mode 1: Operate the I2C timer as two 16-bit timers : Timer Bytes [3 : 2] and Timer Bytes [1 : 0]. 10: I2C Timer Mode 2: Operate the I2C timer as four independent 8-bit timers : Timer Byte 3, Timer Byte 2, Timer Byte 1, and Timer Byte 0. 11: I2C Timer Mode 3: Operate the I2C timer as one 16-bit and two 8-bit timers : Timer Bytes [3 : 2], Timer Byte 1, and Timer Byte 0.

Table 25.3. I2C0_CONFIG Register Bit Descriptions

Bit	Name	Function
27	T3RUN	**I2C Timer Byte 3 Run.** When the I2C Timer is configured for Mode 2, this bit enables the clock to the 8-bit timer formed by Timer Byte 3. This bit has no effect if the I2C module is enabled (I2CEN = 1). 0: Stop Timer Byte 3. 1: Start Timer Byte 3 running.
26	T2RUN	**I2C Timer Byte 2 Run.** When the I2C Timer is configured for Mode 1 or 3, this bit enables the clock to the 16-bit timer formed by Timer Bytes [3 : 2]. In Mode 2, this bit enables the clock to the 8-bit timer formed by Timer Byte 2. This bit has no effect if the I2C module is enabled (I2CEN = 1). 0: Stop Timer Byte 2. 1: Start Timer Byte 2 running.
25	T1RUN	**I2C Timer Byte 1 Run.** When the I2C Timer is configured for Mode 2 or 3, this bit enables the clock to the 8-bit timer formed by Timer Byte 1. This bit has no effect if the I2C module is enabled (I2CEN = 1). 0: Stop Timer Byte 1. 1: Start Timer Byte 1 running.
24	T0RUN	**I2C Timer Byte 0 Run.** When the I2C Timer is configured for Mode 1 or 3, this bit enables the clock to the 32-bit timer formed by Timer Bytes [3 : 2 : 1 : 0]. In Mode 1, this bit enables the clock to the 16-bit timer formed by Timer Bytes [1 : 0]. In Mode 2 or 3, this bit enables the clock to the 8-bit timer formed by Timer Byte 0. This bit has no effect if the I2C module is enabled (I2CEN = 1). 0: Stop Timer Byte 0. 1: Start Timer Byte 0 running.
23:22	BP	**Transfer Byte Pointer.** This field indicates the byte of the current transfer being sent or received. This setting has no effect when using the I2C module with the DMA.
21:20	BC	**Transfer Byte Count.** This field is the number of bytes to transmit or receive when using the I2C module in software mode (DMAEN = 0). This field has no effect when DMA Mode is enabled.
19:18	Reserved	Must write reset value.
17	T3IEN	**I2C Timer Byte 3 Interrupt Enable.** 0: Disable the I2C Timer Byte 3 and SCL low timeout interrupt. 1: Enable the I2C Timer Byte 3 and SCL low timeout interrupt (T3I).
16	T2IEN	**I2C Timer Byte 2 Interrupt Enable.** 0: Disable the I2C Timer Byte 2 interrupt. 1: Enable the I2C Timer Byte 2 interrupt (T2I).

Table 25.3. I2C0_CONFIG Register Bit Descriptions

Bit	Name	Function
15	T1IEN	**I2C Timer Byte 1 Interrupt Enable.** 0: Disable the I2C Timer Byte 1 interrupt. 1: Enable the I2C Timer Byte 1 interrupt (T1I).
14	T0IEN	**I2C Timer Byte 0 Interrupt Enable.** 0: Disable the I2C Timer Byte 0 interrupt. 1: Enable the I2C Timer Byte 0 interrupt (T0I).
13	ARBLIEN	**Arbitration Lost Interrupt Enable.** 0: Disable the arbitration lost interrupt. 1: Enable the arbitration lost interrupt (ARBLI).
12	STAIEN	**Start Interrupt Enable.** 0: Disable the start interrupt. 1: Enable the start interrupt (STAI).
11	TXIEN	**Transmit Done Interrupt Enable.** 0: Disable the transmit done interrupt. 1: Enable the transmit done interrupt (TXI).
10	RXIEN	**Receive Done Interrupt Enable.** 0: Disable the receive done interrupt. 1: Enable the receive done interrupt (RXI).
9	ACKIEN	**Acknowledge Interrupt Enable.** 0: Disable the acknowledge interrupt. 1: Enable the acknowledge interrupt (ACKI).
8	STOIEN	**Stop Interrupt Enable.** 0: Disable the stop interrupt. 1: Enable the stop interrupt (STOI).
7:6	Reserved	Must write reset value.
5:0	SCALER	**I2C Clock Scaler.** The I2C module clock frequency is given by the equation: $F_{I2C} = \frac{F_{APB}}{(64 - SCALER)}$

Register 25.3. I2C0_SADDRESS: Slave Address

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								ADDRESS							Reserved
Type	R					RW			RW							R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_SADDRESS = 0x4000_9020

Table 25.4. I2C0_SADDRESS Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:1	ADDRESS	**Slave Address.** This field contains the 7-bit Slave Address. If slave modes are enabled, the slave will respond with an ACK to any incoming address that matches ADDRESS after being filtered by MASK.
0	Reserved	Must write reset value.

Register 25.4. I2C0_SMASK: Slave Address Mask

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								MASK							Reserved
Type	R					RW			RW							R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_SMASK = 0x4000_9030

Table 25.5. I2C0_SMASK Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:1	MASK	**Slave Address Mask.** This field contains the 7-bit Slave Address Mask. If slave modes are enabled, the slave will respond with an ACK to any incoming address that matches ADDRESS after being filtered by MASK. Any bits set to 1 in MASK will result in the corresponding bit in the incoming address comparing to ADDRESS.
0	Reserved	Must write reset value.

Register 25.5. I2C0_DATA: Data Buffer Access

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_DATA = 0x4000_9040

Table 25.6. I2C0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**Data.** This field contains the four byte I2C transmit and receive buffer. For each transaction, the least significant byte will be sent or received first.

Register 25.6. I2C0_TIMER: Timer Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	T3								T2							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1								T0							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_TIMER = 0x4000_9050

Table 25.7. I2C0_TIMER Register Bit Descriptions

Bit	Name	Function
31:24	T3	**Timer Byte 3.** If the I2C module is enabled (I2CEN = 1), Timer Byte 3 becomes bits [19:12] of the SCL low timeout counter. If the I2C module is disabled (I2CEN = 0), Timer Byte 3 is available for general purpose use and can be set to different modes using the TMD bits.
23:16	T2	**Timer Byte 2.** If the I2C module is enabled (I2CEN = 1), Timer Byte 2 becomes bits [11:4] of the SCL low timeout counter. If the I2C module is disabled (I2CEN = 0), Timer Byte 2 is available for general purpose use and can be set to different modes using the TMD bits.
15:8	T1	**Timer Byte 1.** If the I2C module is enabled (I2CEN = 1), Timer Byte 1 is used as the SCL clock high or low period timer. If the I2C module is disabled (I2CEN = 0), Timer Byte 1 is available for general purpose use and can be set to different modes using the TMD bits.
7:0	T0	**Timer Byte 0.** If the I2C module is enabled (I2CEN = 1), Timer Byte 0 is used as the SDA data setup or hold time period and the SCL free timeout period. If the I2C module is disabled (I2CEN = 0), Timer Byte 0 is available for general purpose use and can be set to different modes using the TMD bits. When used for I2C operations, T0 will automatically be reloaded by hardware from either the T0RL, HOLD, or SETUP fields as necessary.

Register 25.7. I2C0_TIMERRL: Timer Reload Values

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	T3RL								T2RL							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	T1RL								T0RL							
Type	RW								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_TIMERRL = 0x4000_9060

Table 25.8. I2C0_TIMERRL Register Bit Descriptions

Bit	Name	Function
31:24	T3RL	**Timer Byte 3 Reload / SCL Low Timeout Bits [19:12].** These bits contain the reload value for I2C Timer Byte 3. Upon a reload event, this value will be latched into the I2C Timer Byte 3 location. When I2C is enabled, these bits are part of the equation for SCL timeout: $T_{SCL\_TO} = \frac{2^{20} - (16 \times [T3RL : T2RL])}{F_{I2C}}$
23:16	T2RL	**Timer Byte 2 Reload / SCL Low Timeout Bits [11:4].** These bits contain the reload value for I2C Timer Byte 2. Upon a reload event, this value will be latched into the I2C Timer Byte 2 location. When I2C is enabled, these bits are part of the equation for SCL timeout: $T_{SCL\_TO} = \frac{2^{20} - (16 \times [T3RL : T2RL])}{F_{I2C}}$

Table 25.8. I2C0_TIMERRL Register Bit Descriptions

Bit	Name	Function
15:8	T1RL	**Timer Byte 1 Reload / SCL High Time.** These bits contain the reload value for I2C Timer Byte 1. Upon a reload event, this value will be latched into the I2C Timer Byte 1 location. When I2C is enabled, these bits dictate the SCL high time, according to the following equation: $T_{SCL\_HIGH} = \frac{256 - T1RL}{F_{I2C}}$
7:0	T0RL	**Timer Byte 0 Reload / Bus Free Timeout.** These bits contain the reload value for I2C Timer Byte 0. Upon a reload event, this value will be latched into the I2C Timer Byte 0 location. When I2C is enabled, these bits dictate the bus free timeout, according to the following equation: $T_{BUS\_FREE} = \frac{256 - T0RL}{F_{I2C}}$

Register 25.8. I2C0_SCONFIG: SCL Signal Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved												SCLLTIMER			
Type	R												R			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SCLL								HOLD				SETUP			
Type	RW								RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_SCONFIG = 0x4000_9070

Table 25.9. I2C0_SCONFIG Register Bit Descriptions

Bit	Name	Function
31:20	Reserved	Must write reset value.
19:16	SCLLTIMER	**SCL Low Timeout Bits [3:0].** When the I2C module is enabled (I2CEN = 1), this read-only field (bits [3:0]) combines with I2C Timer Byte 3 (bits [19:12]) and I2C Timer Byte 2 (bits [11:4]) to create a 20-bit SCL low timeout counter.
15:8	SCLL	**SCL Low Time.** This field provides the I2C Timer Byte 1 reload value used to generate the SCL low time. This is given by the equation: $T_{SCL\_LOW} = \frac{256 - SCLL}{F_{I2C}}$
7:4	HOLD	**Data Hold Time Extension.** This field provides an alternate I2C Timer Byte 0 reload value to extend the data hold time. The additional hold time is given by the following equation: $T_{HOLD} = \frac{16 - HOLD}{F_{I2C}}$ Note : When HOLD = 0, no additional hold time is added.

Table 25.9. I2C0_SCONFIG Register Bit Descriptions

Bit	Name	Function
3:0	SETUP	**Data Setup Time Extension.** This field provides an alternate I2C Timer Byte 0 reload value to extend data setup time. This is given by the equation: $T_{SETUP} = \frac{17 - SETUP}{F_{I2C}}$ Note : When SETUP = 0, the setup time is reduced to a single APB clock.

Register 25.9. I2C0_I2CDMA: DMA Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMAEN	Reserved														
Type	RW	R														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								DMALEN							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

I2C0_I2CDMA = 0x4000_9080

Table 25.10. I2C0_I2CDMA Register Bit Descriptions

Bit	Name	Function
31	DMAEN	**DMA Mode Enable.** 0: Disable I2C DMA data requests. 1: Enable I2C DMA data requests.
30:8	Reserved	Must write reset value.
7:0	DMALEN	**DMA Transfer Length.** If DMAEN is set to 1, this field indicates the number of bytes to transfer with the I2C module. When DMA operations are enabled, DMALEN = 0, and TXARM or RXARM is set to 1, the module will transfer or receive data indefinitely until firmware clears TXARM or RXARM.

25.10. I2C0 Register Memory Map

Table 25.11. I2C0 Memory Map

I2C0_DATA	I2C0_SMASK	I2C0_SADDRESS	I2C0_CONFIG	I2C0_CONTROL	Register Name
0x4000_9040	0x4000_9030	0x4000_9020	0x4000_9010	0x4000_9000	ALL Address
ALL	ALL	ALL	ALL | SET | CLR	ALL | SET | CLR	Access Methods
DATA			TIMEREN	I2CEN	Bit 31
			Reserved	RESET	Bit 30
			TMD	GCEN	Bit 29
				Reserved	Bit 28
			T3RUN	LBACKEN	Bit 27
			T2RUN	Reserved	Bit 26
			T1RUN	HACKEN	Bit 25
			T0RUN	SMINH	Bit 24
			BP	DBGMD	Bit 23
				FMD	Bit 22
			BC	ATXRXEN	Bit 21
				SLVAF	Bit 20
			Reserved	TXARM	Bit 19
				RXARM	Bit 18
			T3IEN	T3I	Bit 17
			T2IEN	T2I	Bit 16
			T1IEN	T1I	Bit 15
			T0IEN	T0I	Bit 14
			ARBLIEN	ARBLI	Bit 13
			STAIEN	STAI	Bit 12
			TXIEN	TXI	Bit 11
			RXIEN	RXI	Bit 10
			ACKIEN	ACKI	Bit 9
			STOIEN	STOI	Bit 8
			Reserved	MSMDF	Bit 7
			SCALER	TXMDF	Bit 6
				STA	Bit 5
				STO	Bit 4
				ACKRQF	Bit 3
				ARBLF	Bit 2
				ACK	Bit 1
				BUSYF	Bit 0
	Reserved	Reserved			
	MASK	ADDRESS			
		Reserved			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 25.11. I2C0 Memory Map

Register Name	I2C0_I2CDMA	I2C0_SCONFIG	I2C0_TIMERRL	I2C0_TIMER
ALL Address	0x4000_9080	0x4000_9070	0x4000_9060	0x4000_9050
Access Methods	ALL	ALL	ALL	ALL
Bit 31	DMAEN	Reserved	T3RL	T3
Bit 30	Reserved			
Bit 29				
Bit 28				
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23			T2RL	T2
Bit 22				
Bit 21				
Bit 20				
Bit 19		SCLLTIMER		
Bit 18				
Bit 17				
Bit 16				
Bit 15		SCLL	T1RL	T1
Bit 14				
Bit 13				
Bit 12				
Bit 11				
Bit 10				
Bit 9				
Bit 8				
Bit 7	DMALEN	HOLD	T0RL	T0
Bit 6				
Bit 5				
Bit 4				
Bit 3		SETUP		
Bit 2				
Bit 1				
Bit 0				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

26. Current Mode Digital-to-Analog Converter (IDAC0)

This section describes the Current Mode DAC (IDAC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the IDAC block, which is used by IDAC0 on all device families covered in this document.

26.1. IDAC Features

The IDAC takes a digital value as an input and outputs a proportional constant current on a pin. The IDAC module includes the following features:

- 10-bit current DAC with output update trigger source options.
- Ability to update on rising, falling, or both edge for any of the external I/O trigger sources (DACnTx).
- Support for three full-scale output modes: 0.5, 1.0, and 2.0 mA.
- Four-word FIFO to aid with high-speed waveform generation or DMA interactions.
- FIFO supports wrapping mode, allowing the four values to be continuously cycled through to achieve 12-bit resolution.
- Individual FIFO overrun, underrun, and went-empty interrupt status sources.
- Support for multiple data packing formats, including: single 10-bit sample per word, dual 10-bit samples per word, or four 8-bit samples per word.
- Support for left- and right-justified data.

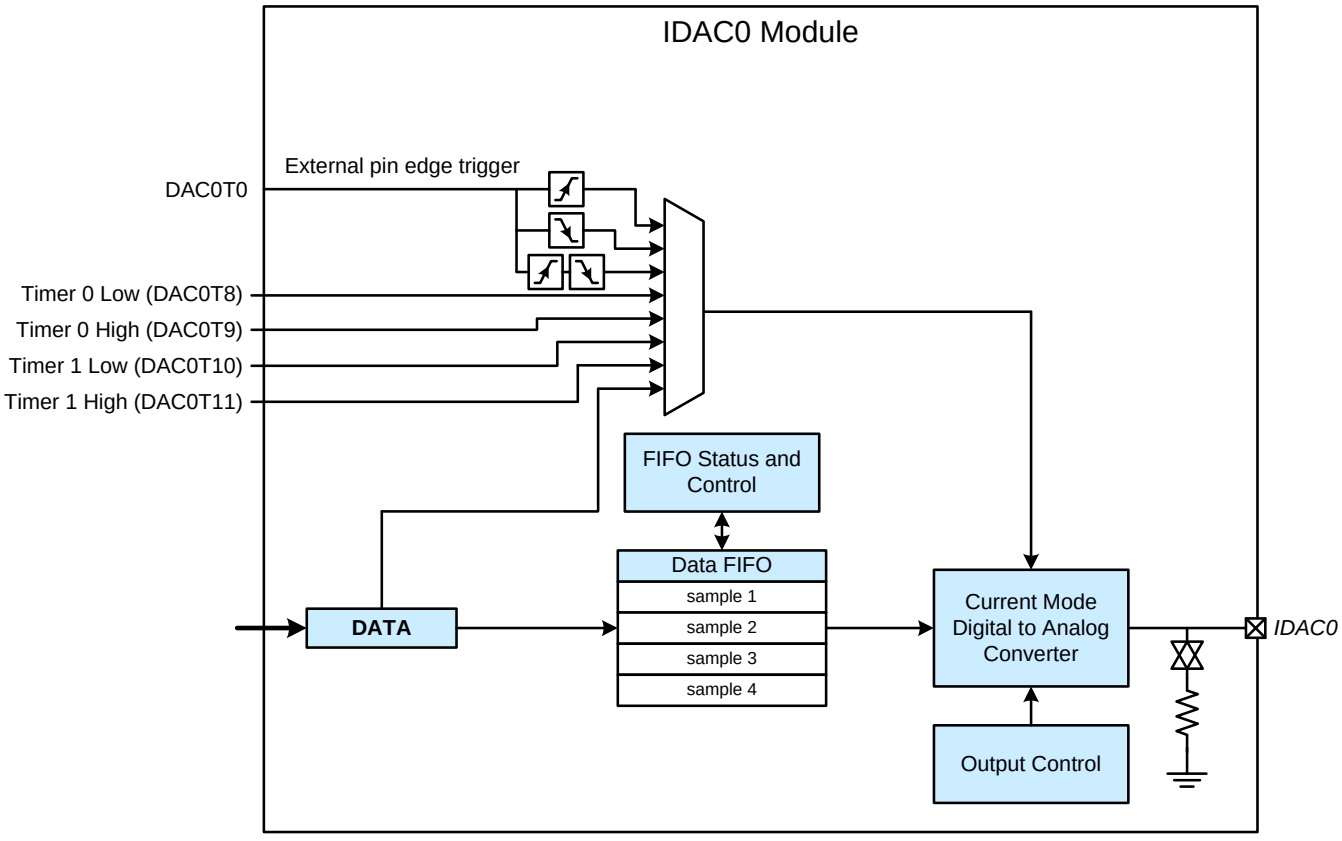


Figure 26.1. IDAC Block Diagram

26.2. IDAC Setup

The various IDAC features and modes are enabled using the CONTROL register. Table 26.2 summarizes the CONTROL register settings for using the IDAC in four different modes: on-demand mode, periodic FIFO wrap mode, periodic FIFO-only mode, and periodic DMA mode. For more detailed bit descriptions, please refer to the IDACn_CONTROL register description.

The entire IDAC block is enabled by setting the IDACEN bit, and can be shut off completely by clearing IDACEN. In most applications, the CONTROL register can be configured with a single write to both configure and enable the IDAC peripheral.

26.2.1. Full Scale Output

The IDAC full scale current output is configured using the OUTMD bit field. Three selectable ranges are available. These are nominally 2.046, 1.023 and 0.5115 mA. The resulting 10-bit LSB sizes of the IDAC in these three modes are 2 µA, 1 µA and 500 nA, respectively.

Optionally, an on-chip load resistor can be enabled by setting the LOADEN bit. This enables an impedance path to ground which effectively produces a voltage at the output pin. The nominal value of the load resistor is given in the device data sheet electrical specification tables. Note that any additional load impedance on the IDAC output pin will affect the output voltage in this mode.

This on-chip load resistor may be useful in some low-accuracy applications. The resistor exhibits a voltage dependence that may contribute some integral non-linearity in addition to the INL of the IDAC itself. The voltage output may also include noise or offset error due to differences in voltage between the internal ground of the resistor and the external board ground. For these reasons, an external resistor is recommended for generation of an accurate ground-referenced voltage on the IDAC output.

26.2.2. Data Format

The IDAC output data can be packed into 32-bit words in three different ways, selected by the INFMT field: single 10-bit data entry, two 10-bit data entries, and four 8-bit data entries. Only single 10-bit data mode is supported when the trigger source is set to on-demand. Note that in 8-bit mode, only the upper 8 bits of the IDAC are used. When the packing format is configured for one of the two 10-bit options, the justification within each 16-bit half word of the 32-bit words can be specified using the JSEL bit. Data will be interpreted as right-justified when JSEL is 0, and left-justified when JSEL is 1. In the 8-bit packing mode, the JSEL bit has no effect. Figure 26.2 shows the data packing mode and justification options.

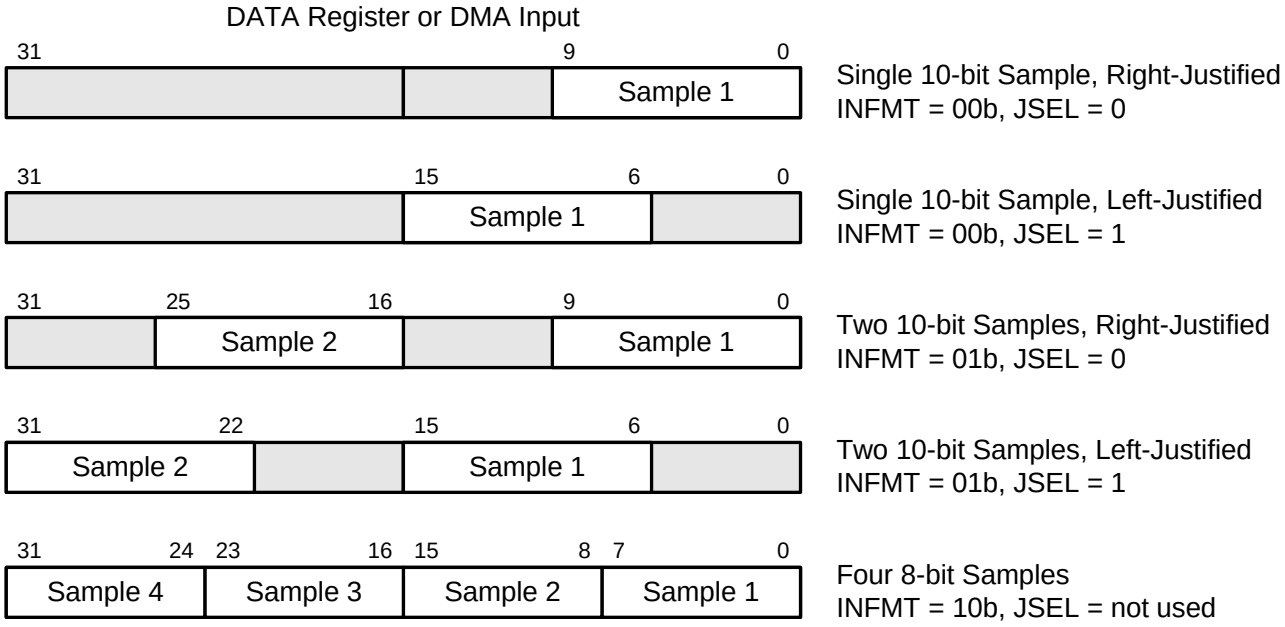


Figure 26.2. Data Packing Modes and Data Justification

26.2.3. Conversion Triggers

IDAC conversions can be triggered “on-demand” with a write to the DATA register, or periodically using one of the internal timer options or external conversion trigger inputs. The DAC0T0 external trigger input can be routed through the crossbar to many of the GPIO signals on the device. The various trigger source options are detailed in Table 26.1.

Two fields in the CONTROL register configure the IDAC trigger source: OUPDT and ETRIG. The OUPDT field selects the trigger event from internal trigger sources (DACnT8 through DACnT11), on-demand mode, or one of three external trigger edge options. When set to an external trigger option, the IDAC can be triggered on the rising edge, falling edge, or both edges of the signal selected by ETRIG. On the SiM3L1xx device family, the only external trigger option is DAC0T0, which is routed to a port I/O pin using the crossbar.

Triggering of the IDAC can be inhibited at any time from firmware by writing the TRIGINH bit to 1. Any trigger events are ignored by the IDAC until the TRIGINH bit is cleared to 0 (except in on-demand mode, where TRIGINH is not used).

Table 26.1. IDAC0 Output Update Triggers

IDAC0 Trigger	Trigger Event
DAC0T8	Timer 0 Low overflow
DAC0T9	Timer 0 High overflow
DAC0T10	Timer 1 Low overflow
DAC0T11	Timer 1 High overflow
DAC0T12	DAC0T0 rising edge
DAC0T13	DAC0T0 falling edge
DAC0T14	DAC0T0 any edge
DAC0T15	“On Demand” by writing to the DATA field

Table 26.2. IDAC Configuration Quick-Reference

Bit Field in CONTROL Register	Operational Mode			
	On-Demand	**Periodic FIFO Wrap (no DMA)**	**Periodic FIFO-Only (no DMA)**	**Periodic with DMA**
IDACEN	1 = Enable IDAC			
DMARUN	0 = Disable DMA			1 = Enable DMA
INFMT	00b = Single, 10-bit Sample	Any Option		
OUPDT	111b = Trigger On-Demand	Any Option Except 111b		
LOADEN	Load resistor enable = Set to 1 if internal load path is desired			
DBGMD	Debug Mode = Set to 1 to let IDAC continue running in debug halt			
JSEL	Data Justification for 10-bit Input Formats: 0 = Right-justify data, 1 = Left-justify data			
OUTMD	Load resistor enable = Set on/off according to application needs			
WEIEN	N/A		Set to enable FIFO Went Empty Interrupt	
URIEN	N/A		Set to enable FIFO Underrun Interrupt	
ORIEN	N/A		Set to enable FIFO Overrun Interrupt	
WRAPEN	N/A	1 = Enable Wrap	0 = Disable Wrap	
TRIGINH	N/A	Set to inhibit IDAC triggering		
BUFRESET	N/A	Set to reset input FIFO		
ETRIG	N/A	Selects external trigger source (if used)		

26.3. Using the IDAC in On-Demand Mode

On-demand mode is useful primarily in DC applications where the IDAC output is changed infrequently. In on-demand mode, the IDAC output is taken directly from the DATA register, and any writes to the DATA register will trigger a corresponding change in current at the IDAC output pin.

The only data input mode supported for on-demand operation is single 10-bit mode. Data writes can be left or right-justified. The FIFO and its associated interrupts are not used in this mode.

26.4. Using the IDAC in Periodic FIFO-Only Mode

Periodic FIFO-only mode is useful in applications where the IDAC output needs to be updated at a specific time interval, and frequent software intervention is allowed. In periodic FIFO-only mode, IDAC updates are configured to occur on the selected trigger signal, and the next output value is pulled from a four-sample FIFO buffer associated with the IDAC. Writes to the DATA register from firmware push new data into the FIFO. It is important that the FIFO has enough room for the sample(s) written to the data register. For example, when INFMT is configured for four 8-bit samples, the entire FIFO must be empty before writing to DATA, but when INFMT is configured for single 10-bit samples, only one FIFO entry need be free.

The number of samples currently waiting in the FIFO is reflected in the LEVEL field of the BUFSTATUS register. The contents of the FIFO can be inspected at any time by reading the BUFFER10 and BUFFER32 registers, and the current IDAC output can always be read from the DATA register.

All three interrupt sources can be enabled in periodic FIFO-only mode. Their meanings for this mode are detailed below.

- FIFO Underrun Interrupt (URI): The URI interrupt flag will be set if an IDAC trigger happens and the FIFO buffer level is zero (i.e. when there is no data to pull from the FIFO). The FIFO read pointer and IDAC output will not be updated when an underrun error occurs.
- FIFO Overrun Interrupt (ORI): The ORI interrupt flag will be set if firmware writes to DATA and there is not enough room in the FIFO for the number of samples written. In this case, no data will be written to the FIFO, and the FIFO write pointer will not be updated.
- FIFO Went Empty Interrupt (WEI): The WEI interrupt flag will be set when an IDAC update reads the final byte from the FIFO into the IDAC output latch, and causes the FIFO level to go to zero. This interrupt can be used by firmware to initiate a new write to the FIFO before the next trigger occurs, and avoid an underrun interrupt.

26.5. Using the IDAC in Periodic FIFO Wrap Mode

Periodic FIFO wrap mode is similar to periodic FIFO-only mode in all ways except for the behavior of the FIFO and the IDAC interrupts. In this mode, the IDAC will continuously pull from the four-sample FIFO in a circular fashion, thereby generating a repeating four sample pattern. The underrun and went empty interrupts are masked off in this mode, and will never occur. The overrun interrupt flag is still active in this mode, but is typically not of much use.

This mode can extend the effective resolution of the DAC by using the four words in the buffer to interpolate between 10-bit values. For example, if the FIFO includes three words of value n and one word of value n+1, then the average output value will be n+0.25.

To load a new set of four samples into the FIFO, the following sequence should be followed:

1. Wait for an IDAC trigger to occur.
2. Reset the FIFO by writing 1 to the BUFRESET bit in the CONTROL register.
3. Load the FIFO with the next set of four samples. The first of these samples should be written before the next trigger event occurs to avoid any glitches in the IDAC output.

26.6. Using the IDAC in Periodic DMA Mode (on select IDAC peripherals only)

A DMA channel can be used to offload core resources and transfer data into the IDAC FIFO. When used in periodic DMA mode, the configuration and capabilities of the IDAC are largely the same as those described in periodic FIFO-only mode. The difference in DMA mode is how data is written into the FIFO, as well as the meaning of the interrupt sources.

When a DMA channel is used to write the FIFO buffer, the FIFO logic will work to keep the buffer full. The FIFO level is monitored, and when the level falls below the number of samples encoded per data word (as specified by the INFMT field), a DMA request will be generated. The DMA request will remain pending until the FIFO no longer has room for new transfers. When configured for a single sample per data word (INFMT = 00b), DMA requests are generated when the LEVEL field in BUFSTATUS is less than or equal to 3. For two samples per data word (INFMT = 01b), DMA requests are generated when LEVEL is less than or equal to two, and for four samples per data word (INFMT = 10b), LEVEL must be 0 to initiate a DMA transfer.

For the IDAC module, the DMA should be configured as follows:

- Source size (SRCSIZE) and destination size (DSTSIZE) are 2 for a word transfer.
- The source address increment (SRCAIMD) is 2 for word increments.
- The destination address increment (DSTAIMD) is 3 for non-incrementing mode.
- The NCOUNT value is N – 1, where N is the number of 4-byte words.
- RPOWER = 0 (1 word transfer per transaction).

Once the DMA is configured, writing a 1 to DMAEN will enable the DMA request from the IDAC. The FIFO will continue to be serviced by the DMA until the specified transfer operation is complete.

When the DMA transfer is complete, the FIFO went empty interrupt flag will be asserted, and the DMAEN bit will be cleared to 0 by hardware. If firmware requires continuous operation of the IDAC as this occurs, it must handle the went empty interrupt, reconfigure the DMA and enable DMA transfers before the next trigger source occurs. Alternatively, the DMA DONE interrupt can be serviced by firmware to allow additional time to set up the next DMA transfer.

All three interrupt sources can be enabled in periodic DMA mode, and it is recommended that firmware do so. The meanings of the interrupts for this mode are detailed below.

- FIFO Underrun Interrupt (URI): The URI interrupt flag will be set if an IDAC trigger happens and the FIFO buffer level is zero (i.e. when there is no data to pull from the FIFO). This can occur if the configured DMA transfer has not completed and a new trigger occurs.
- FIFO Overrun Interrupt (ORI): The ORI interrupt flag will be set if a DMA transfer occurs when there is not enough room in the FIFO for the number of samples written. In this case, no data will be written to the FIFO, and the FIFO write pointer will not be updated. An overrun error should not occur when using the DMA unless there is a firmware conflict with the FIFO.
- FIFO Went Empty Interrupt (WEI): In DMA mode, the WEI interrupt flag is only set at the end of a DMA transfer. This enables the WEI interrupt to be used by firmware to initiate the next DMA sequence if needed.

26.7. Adjusting the IDAC Output Current

The output current of the IDAC is factory calibrated to provide the target current for each OUTMD setting. However, the output current can be adjusted slightly in 32 steps using the GAINADJ field. A value of zero in this field represents the minimum current the IDAC can output at the current OUTMD setting, and a maximum value of 31 in this field represents the maximum current. Each step adjusts the output current by about 1.5%.

This GAINADJ field can be used to calibrate small gain errors that result when using either the internal load resistor or a wider tolerance external load resistor. If the device has an ADC module, the IDAC can be connected internally to the ADC if the IDAC output pin is supported on the ADC input mux or connected externally by shorting the IDAC output and ADC input pins together. Firmware can then use the ADC to measure the resulting voltage on the IDAC output and adjust the GAINADJ field until reaching the desired voltage on the IDAC output.

26.8. Debug Mode

Firmware can set the DBGMD bit to force the IDAC module to halt (ignore incoming triggers) on a debug breakpoint. Clearing the DBGMD bit forces the module to continue operating while the core halts in debug mode.

26.9. IDAC0 Registers

This section contains the detailed register descriptions for IDAC0 registers.

Register 26.1. IDAC0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IDACEN	LOADEN	DBGMD	Reserved						WEIEN	URIEN	ORIEN	Reserved			WRAPEN
Type	RW	RW	RW	R						RW	RW	RW	R			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved		TRIGINH	BUFRESET	JSEL	DMARUN	INFMT		OUTMD		ETRIG			OUPDT		
Type	R		RW	W	RW	RW	RW		RW		RW			RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Register ALL Access Address

IDAC0_CONTROL = 0x4003_1000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 26.3. IDAC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	IDACEN	**IDAC Enable.** 0: Disable the IDAC. 1: Enable the IDAC.
30	LOADEN	**Load Resistor Enable.** Enables an on-chip load resistor to ground. 0: Disable the internal load resistor. 1: Enable the internal load resistor.
29	DBGMD	**IDAC Debug Mode.** 0: The IDAC module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the IDAC module to halt (ignore update triggers).
28:23	Reserved	Must write reset value.
22	WEIEN	**FIFO Went Empty Interrupt Enable.** Enables the FIFO went empty interrupt flag (WEI) to generate an IDAC interrupt when set to 1.

Table 26.3. IDAC0_CONTROL Register Bit Descriptions

Bit	Name	Function
21	URIEN	**FIFO Underrun Interrupt Enable.** Enables the FIFO underrun interrupt flag (URI) to generate an IDAC interrupt when set to 1.
20	ORIEN	**FIFO Overrun Interrupt Enable.** Enables the FIFO overrun interrupt flag (ORI) to generate an IDAC interrupt when set to 1.
19:17	Reserved	Must write reset value.
16	WRAPEN	**Wrap Mode Enable.** Enables IDAC to repeatedly cycle over the FIFO contents in a circular fashion. 0: The IDAC will not wrap when it reaches the end of the data buffer. 1: The IDAC will cycle through the data buffer contents.
15:14	Reserved	Must write reset value.
13	TRIGINH	**Trigger Source Inhibit.** Setting this bit to 1 will mask of any periodic trigger sources. No updates to the IDAC will occur when this bit is set, unless the IDAC is configured for on-demand mode. When cleared to 0, IDAC updates will resume on the next trigger source (trigger sources are not queued).
12	BUFRESET	**Data Buffer Reset.** Writing a 1 to this bit resets the data buffer. Writing a 0 has no effect, and this bit always reads back as 0.
11	JSEL	**Data Justification Select.** This bit selects the data justification in 10-bit input modes. 0: Data is right-justified. 1: Data is left-justified.
10	DMARUN	**DMA Run.** Writing a 1 to this bit enables DMA transfers. This bit is automatically cleared when DMA operations are complete.
9:8	INFMT	**Data Input Format.** This field determines the interpretation of data written to the IDAC. Only single, 10-bit samples are supported in on-demand mode. For periodic FIFO-only mode or periodic FIFO wrap mode, FIFO writes occur on a write to the DATA register. In DMA mode, FIFO writes occur on a DMA event. 00: Writes are interpreted as one 10-bit sample. 01: Writes are interpreted as two 10-bit samples. 10: Writes are interpreted as four 8-bit samples. 11: Reserved.
7:6	OUTMD	**Output Mode.** This field selects the IDAC full-scale output current. 00: The full-scale output current is 0.5 mA. 01: The full-scale output current is 1 mA. 10: The full-scale output current is 2 mA. 11: Reserved.

Table 26.3. IDAC0_CONTROL Register Bit Descriptions

Bit	Name	Function
5:3	ETRIG	**Edge Trigger Source Select.** When OUPDT is configured for any of the edge-trigger options (100b, 101b, or 110b), this field selects the specific external trigger source. 000: Select DACnT0 as the IDAC external trigger source. 001: Select DACnT1 as the IDAC external trigger source. 010: Select DACnT2 as the IDAC external trigger source. 011: Select DACnT3 as the IDAC external trigger source. 100: Select DACnT4 as the IDAC external trigger source. 101: Select DACnT5 as the IDAC external trigger source. 110: Select DACnT6 as the IDAC external trigger source. 111: Select DACnT7 as the IDAC external trigger source.
2:0	OUPDT	**Output Update Trigger.** This field selects the trigger source for IDAC output updates. 000: The IDAC output updates using the DACnT8 trigger source. 001: The IDAC output updates using the DACnT9 trigger source. 010: The IDAC output updates using the DACnT10 trigger source. 011: The IDAC output updates using the DACnT11 trigger source. 100: The IDAC output updates on the rising edge of the trigger source selected by ETRIG. 101: The IDAC output updates on the falling edge of the trigger source selected by ETRIG. 110: The IDAC output updates on any edge of the trigger source selected by ETRIG. 111: The IDAC output updates on write to DATA register (On Demand).

Register 26.2. IDAC0_DATA: Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

IDAC0_DATA = 0x4003_1010

Table 26.4. IDAC0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**Output Data.** When the OUPDT field is set to On-Demand mode, writes to this register update the IDAC value immediately, and are assumed to contain a single 10-bit sample. For all other trigger sources, writes to this register are pushed into the data buffer in the format specified by the INFMT field. Reads from this register always return the current output value in the IDAC latch.

Register 26.3. IDAC0_BUFSTATUS: FIFO Buffer Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									WEI	URI	ORI	Reserved	LEVEL		
Type	R									RW	RW	RW	R	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

IDAC0_BUFSTATUS = 0x4003_1020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 26.5. IDAC0_BUFSTATUS Register Bit Descriptions

Bit	Name	Function
31:7	Reserved	Must write reset value.
6	WEI	**FIFO Went Empty Interrupt Flag.** This bit is set to 1 by hardware when the last sample is transferred from the FIFO into the IDAC output latch. This bit must be cleared by software.
5	URI	**FIFO Underrun Interrupt Flag.** This bit is set to 1 by hardware when a FIFO underrun has occurred. This bit must be cleared by software.
4	ORI	**FIFO Overrun Interrupt Flag.** This bit is set to 1 by hardware when a FIFO overrun has occurred. This bit must be cleared by software.
3	Reserved	Must write reset value.
2:0	LEVEL	**FIFO Level.** Indicates the number of words currently pending in the output data FIFO. 000: The data FIFO is empty. 001: The data FIFO contains one word. 010: The data FIFO contains two words. 011: The data FIFO contains three words. 100: The data FIFO is full and contains four words. 101-111: Reserved.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 26.4. IDAC0_BUFFER10: FIFO Buffer Entries 0 and 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUFFER1															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BUFFER0															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

IDAC0_BUFFER10 = 0x4003_1030

Table 26.6. IDAC0_BUFFER10 Register Bit Descriptions

Bit	Name	Function
31:16	BUFFER1	**FIFO Buffer Entry 1.** This field is the second pending IDAC output. It is justified according to the JSEL selection.
15:0	BUFFER0	**FIFO Buffer Entry 0.** This field is the first pending IDAC output. It is justified according to the JSEL selection.

Register 26.5. IDAC0_BUFFER32: FIFO Buffer Entries 2 and 3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	BUFFER3															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BUFFER2															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

IDAC0_BUFFER32 = 0x4003_1040

Table 26.7. IDAC0_BUFFER32 Register Bit Descriptions

Bit	Name	Function
31:16	BUFFER3	**FIFO Buffer Entry 3.** This field is the fourth pending IDAC output. It is justified according to the JSEL selection.
15:0	BUFFER2	**FIFO Buffer Entry 2.** This field is the third pending IDAC output. It is justified according to the JSEL selection.

Register 26.6. IDAC0_GAINADJ: Output Current Gain Adjust

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											GAINADJ				
Type	R											RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	X	X	X	X	X

Register ALL Access Address
IDAC0_GAINADJ = 0x4003_1050

Table 26.8. IDAC0_GAINADJ Register Bit Descriptions

Bit	Name	Function
31:5	Reserved	Must write reset value.
4:0	GAINADJ	**Output Current Gain Adjust.** This field is factory calibrated to produce the target full-scale output current for all OUTMD settings. However, firmware can modify this field to adjust the output current of the IDAC up or down. A value of 0 represents the minimum current setting, and a value of 31 represents the maximum current setting. Each step adjusts the output current by about 1.5%.

26.10. IDAC0 Register Memory Map

Table 26.9. IDAC0 Memory Map

Register Name	IDAC0_CONTROL	IDAC0_DATA	IDAC0_BUFSTATUS	IDAC0_BUFFER10
ALL Address	0x4003_1000	0x4003_1010	0x4003_1020	0x4003_1030
Access Methods	ALL | SET | CLR	ALL	ALL | SET | CLR	ALL
Bit 31	IDACEN	DATA	Reserved	BUFFER1
Bit 30	LOADEN			
Bit 29	DBGMD			
Bit 28	Reserved			
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23				
Bit 22	WEIEN			
Bit 21	URIEN			
Bit 20	ORIEN			
Bit 19	Reserved			
Bit 18				
Bit 17				
Bit 16	WRAPEN			
Bit 15	Reserved			BUFFER0
Bit 14				
Bit 13	TRIGINH			
Bit 12	BUFRESET			
Bit 11	JSEL			
Bit 10	DMARUN			
Bit 9	INFMT			
Bit 8				
Bit 7	OUTMD			
Bit 6				
Bit 5	ETRIG		WEI	
Bit 4			URI	
Bit 3			ORI	
Bit 2	OUPDT		Reserved	
Bit 1			LEVEL	
Bit 0				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 26.9. IDAC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IDAC0_BUFFER32	0x4003_1040	ALL	BUFFER3																BUFFER2																
IDAC0_GAINADJ	0x4003_1050	ALL	Reserved																														GAINADJ	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

27. LCD Controller (LCD0)

This section describes the LCD Controller (LCD) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the LCD block, which is used by all device families covered in this document.

27.1. LCD Features

The LCD module includes the following features:

- Up to 40 segment pins and 4 common pins.
- Supports LCDs with 1/2 or 1/3 bias.
- Includes an on-chip charge pump with programmable output that allows firmware to control the contrast independent of the supply voltage.
- The RTC timer clock (RTC0TCLK) determines the LCD timing and refresh rate.
- Independent missing clock detector.
- All LCD waveforms are generated on-chip based on the contents of the LCD0 registers with flexible waveform control.
- LCD segments and common signals can be placed in an intermediate state for a configurable number of RTC clock cycles before switching to the next state to reduce power consumption due to display loading.
- Includes a VBAT monitor that can serve as a wakeup source for Power Mode 8.
- Supports four hardware auto-contrast modes: bypass, constant, minimum, and auto-bypass.
- Supports hardware blinking for up to 8 segments.

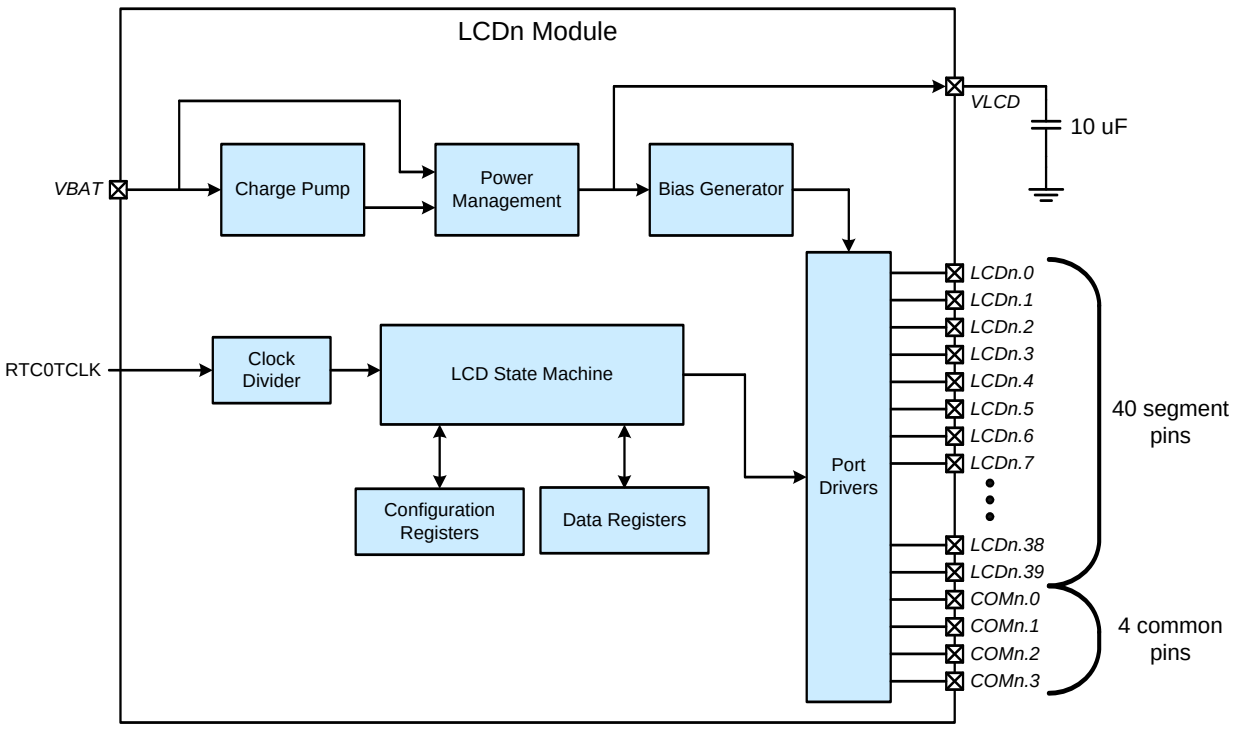


Figure 27.1. LCD0 Block Diagram

27.2. Pin Assignment

The external pin connections for the LCD0 module include up to 4 COM pins and 40 segment pins. Any pins used by the LCD0 module should be configured to analog mode in the PBCFG module and skipped by the crossbar. Refer to the pinout table in the data sheet for specific LCD pin connections.

27.3. Configuring the LCD Segment Driver

The LCD segment driver supports multiple mux options: static, 2-mux, 3-mux, and 4-mux mode. It also supports 1/2 and 1/3 bias options. The desired mux mode and bias are configured through the SEGCONTROL register. A divide value may also be applied to the RTC timer clock (RTC0TCLK) output before being used as the LCD0 clock source.

The following procedure is recommended for using the LCD Segment Driver:

1. Enable the LCDEN bit in CLKCTRL0_APBCLKG0 before writing to LCD registers.
2. Select the desired mux mode (static, 2 mux, 3 mux, or 4 mux) using the SEGMD field in the SEGCONTROL register.
3. Configured the BIASMD field of SEGCONTROL for the desired mux mode. One half bias is used for 2 mux mode. The other mux modes use one third bias.
4. Configure the following bits according to Table 27.1 for the desired auto-contrast mode.
   a. CPFPDEN in CONFIG0
   b. CPBEN in CONFIG0
   c. CPACEN in CONFIG0
   d. VBMEN in VBMCONTROL
5. Configure the RTCCLKDIV bits for the desired clock divider in the CLKCONTROL register.
6. Configure the CLKDIV bits in the CLKCONTROL register for the desired refresh rate.
7. Determine the I/O pins which will be used for the LCD function.
8. Enable the desired segments by writing to the SEGMASK0 and SEGMASK1 registers.
9. Configure the CTRST field in the CTRSTCONTROL register for the desired contrast voltage.
10. Enable MISC0EN bit in CLKCTRL0_APBCLKG1 before writing to RTC registers.
11. Configure RTC as desired.
12. Enable the RTC0TCLK signal by setting the CLKOEN bit in the RTC0_CONFIG register. (This enables RTC timer clock to the LCD module.)
13. Enable PB0CEN bit in CLKCTRL0_APBCLKG1 before writing to PB registers.
14. Configure the relevant PB pins as analog inputs by clearing the corresponding bits in the PBMDSEL register, and setting the corresponding bits in the PB register to 1.
15. Enable the LCD by setting the LCDEN bit in the CONFIG register.

27.4. Mapping Data Registers to LCD Pins

The LCD0 data registers are organized as five 32-bit registers in memory (SEGDATA0-SEGDATA5). Each nibble in these registers controls one LCD output pin. There are 40 nibbles used to control the 40 segment pins.

Each LCD0 segment pin can control 1, 2, 3, or 4 LCD segments depending on the selected mux mode. The least significant bit of each nibble controls the segment connected to the backplane signal COM0.0. The next to least significant bit controls the segment associated with COM0.1, the next bit controls the segment associated with COM0.2, and the most significant bit in the 4-bit nibble controls the segment associated with COM0.3.

In static mode, only the least significant bit in each nibble is used and the three remaining bits in each nibble are ignored. In 2-mux mode, only the two least significant bits are used; in 3-mux mode, only the three least significant bits are used, and in 4-mux mode, each of the 4 bits in the nibble controls one LCD segment. Bits with a value of 1 turn on the associated segment and bits with a value of 0 turn off the associated segment.

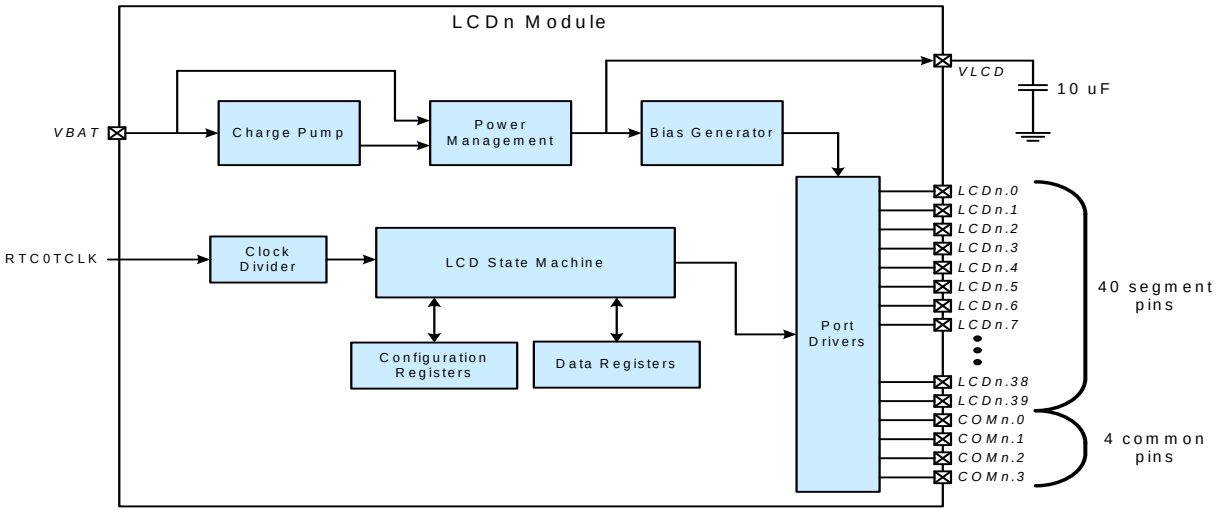


Figure 27.2. LCD Data Register to LCD Pin Mapping

27.5. LCD Contrast Adjustment

The LCD Bias voltages which determine the LCD contrast are generated using the VBAT supply voltage or the on-chip charge pump. There are four contrast control modes to accommodate a wide variety of applications and supply voltages. The target contrast voltage is programmable in 60 mV steps from 1.9 to 3.72 V. The LCD contrast voltage is controlled by the CTRSTCONTROL register and the contrast control mode is selected by setting the appropriate bits in the CONFIG and VBMCONTROL registers. Refer to Table 27.1 for information on the different bit settings for each mode.

Note: An external 10 µF decoupling capacitor is required on the VLCD pin to create a charge reservoir at the output of the charge pump.

Table 27.1. Bit Configurations to select Contrast Control Modes

Contrast Control Mode	HCVLPMEN in CONFIG	HCVCBYPEN in CONFIG	CPACEN in CONFIG	VBMEN in VBMCONTROL
Bypass Mode (contrast disabled)	0	1	0	0
Minimum Contrast Mode	0	1	1	1
Constant Contrast Mode	1*	0	1	1
Auto-Bypass Mode	1*	0	0	1
*** May be set to 0 to support increased load currents.**				

27.5.1. Bypass Mode

In bypass mode, the contrast control circuitry is disabled and the VLCD voltage follows the VBAT supply voltage, as shown in Figure 27.3. This mode is useful in systems where the VBAT voltage always remains constant and will provide the lowest LCD power consumption.

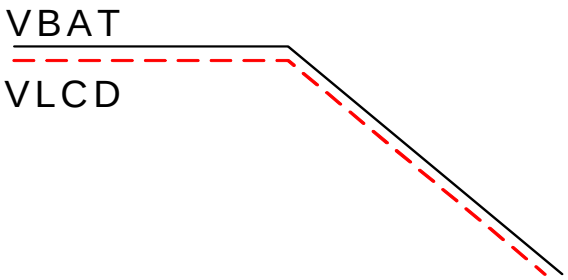


Figure 27.3. Bypass Mode

27.5.2. Minimum Contrast Mode

In minimum contrast mode, a minimum contrast voltage is maintained, as shown in Figure 27.4. The VLCD supply is powered directly from VBAT as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to achieve the desired minimum contrast voltage on VLCD.

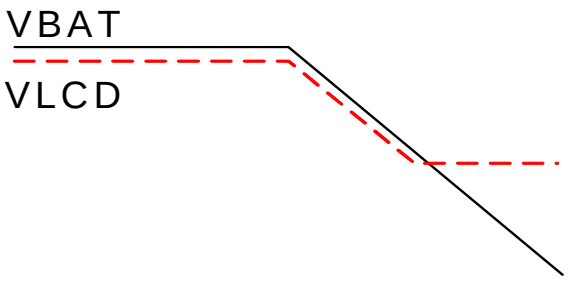


Figure 27.4. Minimum Contrast Mode

27.5.3. Constant Contrast Mode

In constant contrast mode, a constant contrast voltage is maintained. The VLCD supply is regulated to the programmed contrast voltage using a variable resistor between VBAT and VLCD as long as VBAT is higher than the programmable VBAT monitor threshold voltage. As soon as the VBAT supply monitor detects that VBAT has dropped below the programmed value, the charge pump will be automatically enabled in order to achieve the desired contrast voltage on VLCD.

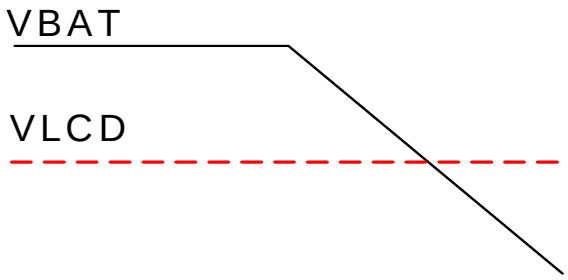


Figure 27.5. Contrast Control Mode 3

27.5.4. Auto-Bypass Mode

In auto-bypass mode, behavior is identical to constant contrast mode as long as VBAT is greater than the VBAT monitor threshold voltage. When VBAT drops below the programmed threshold, the device automatically enters bypass mode, powering VLCD directly from VBAT. The charge pump is always disabled in this mode.

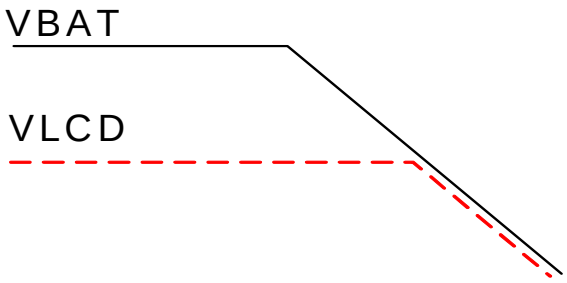


Figure 27.6. Contrast Control Mode 4

27.6. Adjusting the VBAT Monitor Threshold

The VBAT Monitor is used primarily for the contrast control function, to detect when VBAT has fallen below a specific threshold. The VBAT monitor threshold may be set independently of the contrast setting or it may be linked to the contrast setting. When the VBAT monitor threshold is linked to the contrast setting, an offset (in 60 mV steps) may be configured so that the VBAT monitor generates a VBAT low condition prior to VBAT dropping below the programmed contrast voltage. The VBMCONTROL register is used to enable and configure the VBAT Monitor. The VBAT monitor may also be enabled as a wake-up source to wake up the device from Sleep mode to indicate that the battery is getting low.

27.7. Setting the LCD Refresh Rate

The clock to the LCD0 module is derived from the RTC oscillator and may be divided down according to the setting of the RTCCLKDIV field in the CLKCONTROL register. The LCD refresh rate is derived from the LCD0 clock and can be programmed using the CLKDIV field. The LCD mux mode must be taken into account when determining the prescaler value. See the CLKDIV register description for more details. For maximum power savings, choose a slow LCD refresh rate and the minimum LCD0 clock frequency. For the least flicker, choose a fast LCD refresh rate.

27.8. Blinking LCD Segments

The LCD driver supports blinking LCD applications such as clock applications where the ":" separator toggles on and off once per second. If the LCD is only displaying the hours and minutes, then the device only needs to wake up once per minute to update the display. The segment blinking is automatically handled by the LCD0 module.

The BLKCONTROL register can be used to enable blinking on any LCD segment connected to the LCD0.0 or LCD0.1 segment pin. In static mode, a maximum of 2 segments can blink. In 2-mux mode, a maximum of 4 segments can blink; in 3-mux mode, a maximum of 6 segments can blink; and in 4-mux mode, a maximum of 8 segments can blink. The BLKMASK field targets the same LCD segments as the least significant byte of the SEGDATA0 register. If a BLKMASK bit corresponding to an LCD segment is set to 1, then that segment will toggle without any software intervention. The BLKREXP field sets the divide ratio for the blinking interval.

27.9. LCD0 Registers

This section contains the detailed register descriptions for LCD0 registers.

Register 27.1. LCD0_CONFIG: Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	CPCS	Reserved		HCVCBMD	HCVCHMD	HCVCFOEN	HCVCBYPEN	Reserved						FBIASCEN	CPACEN
Type	R	R	R		RW	RW	RW	RW	R		RW				RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	RBGSEN	BIASSEN	CMPBLPEN	CPOLPEN	VBMLPEN	HCVLPMEN	CPBEN	DCDCSTDBYEN	DCDCBIASEN	BIASEN	RTCCEN	MCDEN	CPFPDEN	Reserved	LCDEN
Type	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	W	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_CONFIG = 0x4004_D000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 27.2. LCD0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	CPCS	**High-Contrast-Voltage Comparator Status.** (Valid only when High-Contrast-Voltage Comparator is enabled.) 0: VBAT is greater than VLCD. 1: VLCD is greater than VBAT.
29:28	Reserved	Must write reset value.
27	HCVCBMD	**High-Contrast-Voltage Comparator Bias.** 0: Set the high-contrast-voltage comparator to high bias mode. 1: Set the high-contrast-voltage comparator to low-bias mode. This is the recommended setting.

Table 27.2. LCD0_CONFIG Register Bit Descriptions

Bit	Name	Function
26	HCVCHMD	**High-Contrast-Voltage Comparator Hysteresis.** 0: Set the high-contrast-voltage comparator to high-hysteresis mode. This is the recommended setting. 1: Set the high-contrast-voltage comparator to low-hysteresis mode.
25	HCVCFOEN	**High-Contrast-Voltage Comparator Force On Enable.** 0: Hardware enables the high-contrast-voltage comparator as needed. 1: High-contrast-voltage comparator force on enabled.
24	HCVCBYPEN	**High-Contrast-Voltage Comparator Bypass Enable.** 0: Hardware enables the high-contrast-voltage comparator as needed. 1: High-contrast-voltage comparator in bypass mode.
23:18	Reserved	Must write reset value.
17	FBIASCEN	**Force Bias Continuous Mode Enable.** 0: The bias operates as configured. 1: Force the bias to operate in continuous mode. The bias will cleanly transition from its configuration settings to continuous mode.
16	CPACEN	**Charge-Pump Auto-Contrast Enable.** 0: VLCD continues to track VBAT when VBAT drops below the programmed VLCD value. 1: The module automatically enables the charge pump and maintains the VLCD voltage when VBAT drops below the programmed VBAT monitor level.
15	Reserved	Must write reset value.
14	RBGSEN	**Reference Band Gap Switching Enable.** 0: Disable reference band gap switching mode. 1: Enable reference band gap switching mode.
13	BIASSEN	**Bias Switching Enable.** 0: Disable bias switching. 1: Enable bias switching.
12	CMPBLPEN	**Comparator Buffer Low-Power Enable.** 0: Disable the comparator buffer low-power mode. 1: Enable the comparator buffer low-power mode.
11	CPOLPEN	**Charge-Pump Oscillator Low-Power Enable.** 0: Disable the charge-pump oscillator low-power mode. 1: Enable the charge-pump oscillator low-power mode.
10	VBMLPEN	**VBAT Monitor Low Power Enable.** 0: Disable the LCD VBAT Monitor low power mode. 1: Enable the LCD VBAT Monitor low power mode.
9	HCVLPMEN	**High-Contrast-Voltage Low-Power Mode Enable.** 0: Disable the high-contrast-voltage low-power mode. 1: Enable the high-contrast-voltage low-power mode. This mode reduces power consumption when VLCD is higher than VBAT.

Table 27.2. LCD0_CONFIG Register Bit Descriptions

Bit	Name	Function
8	CPBEN	**Charge Pump Bypass Enable.** 0: The LCD charge pump generates the VLCD voltage. 1: Bypass the LCD charge pump and connect VLCD directly to VBAT.
7	DCDCSTDBYEN	**DCDC Bias Standby Enable.** 0: The DCDC bias is enabled in Power Mode 8. 1: The DCDC bias is disabled in Power Mode 8.
6	DCDCBIASEN	**DCDC Bias Output Enable.** 0: Disable the secondary bias current output. 1: Enable the secondary bias current output.
5	BIASEN	**Bias Enable.** 0: Disable the LCD bias current. 1: Enable the LCD bias current.
4	RTCCEN	**RTC Clock Request Enable.** 0: The LCD module does not require the RTC clock. 1: The LCD module requires an active and valid RTC clock (RTC0TCLK).
3	MCDEN	**LCD Missing Clock Detector Enable.** 0: Disable the dedicated LCD missing clock detector. 1: Enable the dedicated LCD missing clock detector.
2	CPFPDEN	**Charge Pump Full Power Drive Mode Enable.** 0: Disable the LCD charge pump's full power drive mode. The charge pump draws less power but operates with reduced output current capabilities. 1: Enable the LCD charge pump's full output drive mode. The charge pump operates at full power.
1	Reserved	Must write reset value.
0	LCDEN	**Module Enable.** 0: Disable the LCD module. 1: Enable the LCD module.

Register 27.2. LCD0_CLKCONTROL: Clock Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		RTCCLKDIV		Reserved											
Type	R		RW		RW									R		
Reset	0	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved						CLKDIV									
Type	RW						RW									
Reset	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_CLKCONTROL = 0x4004_D020

Table 27.3. LCD0_CLKCONTROL Register Bit Descriptions

Bit	Name	Function
31:30	Reserved	Must write reset value.
29:28	RTCCLKDIV	**RTC Input Clock Divider.** This field selects the RTC clock (RTC0TCLK) divider setting to generate the LCD input clock. The LCD clock should be maintained at approximately 16 kHz. The LCD clock frequency is calculated from the RTC0TCLK frequency as follows: $F_{LCD} = \frac{F_{RTC0TCLK}}{2^{RTCCLKDIV}}$
27:10	Reserved	Must write reset value.
9:0	CLKDIV	**Clock Divider.** This field sets the clock divider value for the LCD refresh rate. The LCD refresh rate is derived from the LCD clock as: $F_{refresh} = \frac{F_{LCD}}{4 \times (SEGMD + 1) \times (CLKDIV + 1)}$

Register 27.3. LCD0_BLKCONTROL: Blinking Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				BLKREXP				BLKMASK							
Type	R				RW				RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_BLKCONTROL = 0x4004_D030

Table 27.4. LCD0_BLKCONTROL Register Bit Descriptions

Bit	Name	Function
31:12	Reserved	Must write reset value.
11:8	BLKREXP	**Hardware Blinking Rate Divider Exponent.** This field sets the blinking rate divider exponent for segments enabled in the BLKMASK field. The minimum setting for this field is 2 and the maximum setting is 13. Settings outside these minimum and maximum values are invalid and should not be used. $F_{blink} = \frac{F_{LCD}}{(CLKDIV + 1) \times 2^{BLKREXP}}$
7:0	BLKMASK	**Hardware Blinking Enable.** This field enables blinking on the lowest eight LCD segments. Setting bit x of this field to 1 enables hardware blinking on segment LCDn.x.

Register 27.4. LCD0_SEGCONTROL: Segment Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved							RPHMD			RPHEN	BLANKEN	Reserved	SEGMD		BIASMD
Type	R							RW			RW	RW	R	RW		RW
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGCONTROL = 0x4004_D040

Table 27.5. LCD0_SEGCONTROL Register Bit Descriptions

Bit	Name	Function
31:9	Reserved	Must write reset value.
8:6	RPHMD	**Reset Phase Mode.** This field controls the duration of the reset phase, when enabled (RPHEN = 1).
5	RPHEN	**Reset Phase Enable.** Setting this bit to 1 enables pin resets between phases for lower power consumption. The RPHMD field sets the duration of this reset phase. 0: Hardware switches the LCD segment and common pin controls directly from one state to another. 1: Hardware switches the LCD segment and common pin controls to intermediate states for several RTC clock cycles before switching to the next state.
4	BLANKEN	**Segment Blank Enable.** 0: Operate segments normally. 1: Blank the LCD by disabling all LCD segment and common pins.
3	Reserved	Must write reset value.
2:1	SEGMD	**Segment Mode.** 00: Select static segment mode with one common COMn.0 used. 01: Select two-mux segment mode with two commons (COMn.0 and COMn.1) used. 10: Select three-mux segment mode with three commons (COMn.0, COMn.1, COMn.2) used. 11: Select four-mux segment mode with four commons (COMn.0, COMn.1, COMn.2 and COMn.3) used.

Table 27.5. LCD0_SEGCONTROL Register Bit Descriptions

Bit	Name	Function
0	BIASMD	**Hardware Bias Mode.** This bit sets the bias mode for dynamic LCD operation. It has no effect in static mode. 0: Select 1/3 bias. Use for three-mux segment mode and four-mux segment mode. 1: Select 1/2 bias. Use for two-mux segment mode.

Register 27.5. LCD0_CTRSTCONTROL: Contrast Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		CPCDEN	Reserved												CTRSTBF
Type	R		RW	RW	R	RW					R					R
Reset	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											CTRST				
Type	R											RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_CTRSTCONTROL = 0x4004_D060

Table 27.6. LCD0_CTRSTCONTROL Register Bit Descriptions

Bit	Name	Function
31:30	Reserved	Must write reset value.
29	CPCDEN	**Charge Pump Capacitor Divider Enable.** 0: Disable the charge pump capacitor divider. 1: Enable the charge pump capacitor divider.
28:17	Reserved	Must write reset value.
16	CTRSTBF	**Contrast Busy Flag.** 0: An update of the internal contrast registers is not in progress. 1: The internal contrast registers are busy updating.
15:5	Reserved	Must write reset value.
4:0	CTRST	**Contrast Voltage.** This field sets the LCD contrast voltage. After setting this field, firmware should poll the contrast busy flag (CTRSTBF) to determine when the update completes.

Register 27.6. LCD0_VBMCONTROL: VBAT Monitor Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VBMEN	VBMOEN	VBMCDEN	Reserved				VBMCLKDIV			Reserved					VBMBF
Type	RW	RW	RW	RW				RW			R					R
Reset	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											VBMTH				
Type	R											RW				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_VBMCONTROL = 0x4004_D070

Table 27.7. LCD0_VBMCONTROL Register Bit Descriptions

Bit	Name	Function
31	VBMEN	**VBAT Monitor Enable.** 0: Disable the VBAT monitor. 1: Enable the VBAT monitor.
30	VBMOEN	**VBAT Monitor Offset Enable.** 0: The VBAT monitor threshold set by the VBMTH field functions as an absolute threshold value for the VBAT monitor. 1: The VBAT monitor threshold set by the VBMTH field functions as an offset to the LCD contrast value set by CTRSTMD.
29	VBMCDEN	**VBAT Monitor Capacitor Divider Enable.** 0: Disable the VBAT monitor capacitor divider. 1: Enable the VBAT monitor capacitor divider.
28:25	Reserved	Must write reset value.
24:22	VBMCLKDIV	**VBAT Monitor Clock Divider.** This field sets the clock divider for the VBAT monitor resistor string.
21:17	Reserved	Must write reset value.
16	VBMBF	**VBAT Monitor Busy Flag.** 0: An update of the internal VBAT monitor registers is not in progress. 1: The internal VBAT monitor registers are busy updating.
15:5	Reserved	Must write reset value.

Table 27.7. LCD0_VBMCONTROL Register Bit Descriptions

Bit	Name	Function
4:0	VBMTH	**VBAT Monitor Threshold.** This field sets the VBAT monitor threshold.

Register 27.7. LCD0_SEGMASK0: Segment Mask 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGEN[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGEN[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGMASK0 = 0x4004_D080

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 27.8. LCD0_SEGMASK0 Register Bit Descriptions

Bit	Name	Function
31:0	SEGEN	**Segment Enable.** Setting a bit to 1 enables the corresponding LCD segment. Bit x of this field corresponds to segment LCDn.x.

Register 27.8. LCD0_SEGMASK1: Segment Mask 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								SEGEN							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGMASK1 = 0x4004_D090

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 27.9. LCD0_SEGMASK1 Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:0	SEGEN	**Segment Enable.** Setting a bit to 1 enables the corresponding LCD segment. Bit 0 of this field corresponds to segment LCDn.32, and bit 7 of this field corresponds to segment LCDn.39.

Register 27.9. LCD0_SEGDATA0: Segment Data 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGPIN7				SEGPIN6				SEGPIN5				SEGPIN4			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGPIN3				SEGPIN2				SEGPIN1				SEGPIN0			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGDATA0 = 0x4004_D0A0

Table 27.10. LCD0_SEGDATA0 Register Bit Descriptions

Bit	Name	Function
31:28	SEGPIN7	**Segment LCDn.7 Control.** This field controls segment LCDn.7. Each nibble in the SEGDATA0 register controls a different segment of the LCD. Bit 0 of the nibble controls the segment value for common COMn.0, bit 1 for common COMn.1, bit 2 for common COMn.2, and bit 3 for common COMn.3.
27:24	SEGPIN6	**Segment LCDn.6 Control.** This field controls segment LCDn.6.
23:20	SEGPIN5	**Segment LCDn.5 Control.** This field controls segment LCDn.5.
19:16	SEGPIN4	**Segment LCDn.4 Control.** This field controls segment LCDn.4.
15:12	SEGPIN3	**Segment LCDn.3 Control.** This field controls segment LCDn.3.
11:8	SEGPIN2	**Segment LCDn.2 Control.** This field controls segment LCDn.2.
7:4	SEGPIN1	**Segment LCDn.1 Control.** This field controls segment LCDn.1.
3:0	SEGPIN0	**Segment LCDn.0 Control.** This field controls segment LCDn.0.

Register 27.10. LCD0_SEGDATA1: Segment Data 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGPIN15				SEGPIN14				SEGPIN13				SEGPIN12			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGPIN11				SEGPIN10				SEGPIN9				SEGPIN8			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGDATA1 = 0x4004_D0B0

Table 27.11. LCD0_SEGDATA1 Register Bit Descriptions

Bit	Name	Function
31:28	SEGPIN15	**Segment LCDn.15 Control.** This field controls segment LCDn.15. Each nibble in the SEGDATA1 register controls a different segment of the LCD. Bit 0 of the nibble controls the segment value for common COMn.0, bit 1 for common COMn.1, bit 2 for common COMn.2, and bit 3 for common COMn.3.
27:24	SEGPIN14	**Segment LCDn.14 Control.** This field controls segment LCDn.14.
23:20	SEGPIN13	**Segment LCDn.13 Control.** This field controls segment LCDn.13.
19:16	SEGPIN12	**Segment LCDn.12 Control.** This field controls segment LCDn.12.
15:12	SEGPIN11	**Segment LCDn.11 Control.** This field controls segment LCDn.11.
11:8	SEGPIN10	**Segment LCDn.10 Control.** This field controls segment LCDn.10.
7:4	SEGPIN9	**Segment LCDn.9 Control.** This field controls segment LCDn.9.
3:0	SEGPIN8	**Segment LCDn.8 Control.** This field controls segment LCDn.8.

Register 27.11. LCD0_SEGDATA2: Segment Data 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGPIN23				SEGPIN22				SEGPIN21				SEGPIN20			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGPIN19				SEGPIN18				SEGPIN17				SEGPIN16			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGDATA2 = 0x4004_D0C0

Table 27.12. LCD0_SEGDATA2 Register Bit Descriptions

Bit	Name	Function
31:28	SEGPIN23	**Segment LCDn.23 Control.** This field controls segment LCDn.23. Each nibble in the SEGDATA2 register controls a different segment of the LCD. Bit 0 of the nibble controls the segment value for common COMn.0, bit 1 for common COMn.1, bit 2 for common COMn.2, and bit 3 for common COMn.3.
27:24	SEGPIN22	**Segment LCDn.22 Control.** This field controls segment LCDn.22.
23:20	SEGPIN21	**Segment LCDn.21 Control.** This field controls segment LCDn.21.
19:16	SEGPIN20	**Segment LCDn.20 Control.** This field controls segment LCDn.20.
15:12	SEGPIN19	**Segment LCDn.19 Control.** This field controls segment LCDn.19.
11:8	SEGPIN18	**Segment LCDn.18 Control.** This field controls segment LCDn.18.
7:4	SEGPIN17	**Segment LCDn.17 Control.** This field controls segment LCDn.17.
3:0	SEGPIN16	**Segment LCDn.16 Control.** This field controls segment LCDn.16.

Register 27.12. LCD0_SEGDATA3: Segment Data 3

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGPIN31				SEGPIN30				SEGPIN29				SEGPIN28			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGPIN27				SEGPIN26				SEGPIN25				SEGPIN24			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGDATA3 = 0x4004_D0D0

Table 27.13. LCD0_SEGDATA3 Register Bit Descriptions

Bit	Name	Function
31:28	SEGPIN31	**Segment LCDn.31 Control.** This field controls segment LCDn.31. Each nibble in the SEGDATA3 register controls a different segment of the LCD. Bit 0 of the nibble controls the segment value for common COMn.0, bit 1 for common COMn.1, bit 2 for common COMn.2, and bit 3 for common COMn.3.
27:24	SEGPIN30	**Segment LCDn.30 Control.** This field controls segment LCDn.30.
23:20	SEGPIN29	**Segment LCDn.29 Control.** This field controls segment LCDn.29.
19:16	SEGPIN28	**Segment LCDn.28 Control.** This field controls segment LCDn.28.
15:12	SEGPIN27	**Segment LCDn.27 Control.** This field controls segment LCDn.27.
11:8	SEGPIN26	**Segment LCDn.26 Control.** This field controls segment LCDn.26.
7:4	SEGPIN25	**Segment LCDn.25 Control.** This field controls segment LCDn.25.
3:0	SEGPIN24	**Segment LCDn.24 Control.** This field controls segment LCDn.24.

Register 27.13. LCD0_SEGDATA4: Segment Data 4

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEGPIN39				SEGPIN38				SEGPIN37				SEGPIN36			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SEGPIN35				SEGPIN34				SEGPIN33				SEGPIN32			
Type	RW				RW				RW				RW			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LCD0_SEGDATA4 = 0x4004_D0E0

Table 27.14. LCD0_SEGDATA4 Register Bit Descriptions

Bit	Name	Function
31:28	SEGPIN39	**Segment LCDn.39 Control.** This field controls segment LCDn.39. Each nibble in the SEGDATA4 register controls a different segment of the LCD. Bit 0 of the nibble controls the segment value for common COMn.0, bit 1 for common COMn.1, bit 2 for common COMn.2, and bit 3 for common COMn.3.
27:24	SEGPIN38	**Segment LCDn.38 Control.** This field controls segment LCDn.38.
23:20	SEGPIN37	**Segment LCDn.37 Control.** This field controls segment LCDn.37.
19:16	SEGPIN36	**Segment LCDn.36 Control.** This field controls segment LCDn.36.
15:12	SEGPIN35	**Segment LCDn.35 Control.** This field controls segment LCDn.35.
11:8	SEGPIN34	**Segment LCDn.34 Control.** This field controls segment LCDn.34.
7:4	SEGPIN33	**Segment LCDn.33 Control.** This field controls segment LCDn.33.
3:0	SEGPIN32	**Segment LCDn.32 Control.** This field controls segment LCDn.32.

27.10. LCD0 Register Memory Map

Table 27.15. LCD0 Memory Map

Register Name	LCD0_CONFIG	LCD0_CLKCONTROL	LCD0_BLKCONTROL	LCD0_SEGCONTROL
ALL Address	0x4004_D000	0x4004_D020	0x4004_D030	0x4004_D040
Access Methods	ALL | SET | CLR	ALL	ALL	ALL
Bit 31	Reserved	Reserved	Reserved	Reserved
Bit 30	CPCS	RTCCLKDIV		
Bit 29	Reserved			
Bit 28				
Bit 27	HCVCBMD	Reserved		
Bit 26	HCVCHMD			
Bit 25	HCVCFOEN			
Bit 24	HCVCBYPEN			
Bit 23	Reserved			
Bit 22				
Bit 21				
Bit 20				
Bit 19				
Bit 18				
Bit 17	FBIASCEN			
Bit 16	CPACEN			
Bit 15	Reserved			
Bit 14	RBGSEN			
Bit 13	BIASSEN			
Bit 12	CMPBLPEN			
Bit 11	CPOLPEN		BLKREXP	
Bit 10	VBMLPEN			
Bit 9	HCVLPMEN	CLKDIV		
Bit 8	CPBEN			
Bit 7	DCDCSTDBYEN		BLKMASK	
Bit 6	DCDCBIASEN			
Bit 5	BIASEN			RPHMD
Bit 4	RTCCEN			RPHEN
Bit 3	MCDEN			BLANKEN
Bit 2	CPFPDEN			Reserved
Bit 1	Reserved			SEGMD
Bit 0	LCDEN			BIASMD

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 27.15. LCD0 Memory Map

Register Name	LCD0_SEGMASK1	LCD0_SEGMASK0	LCD0_VBMCONTROL	LCD0_CTRSTCONTROL
ALL Address	0x4004_D090	0x4004_D080	0x4004_D070	0x4004_D060
Access Methods	ALL | SET | CLR	ALL | SET | CLR	ALL	ALL
Bit 31	Reserved	SEGEN	VBMEN	Reserved
Bit 30			VBMOEN	
Bit 29			VBMCDEN	CPCDEN
Bit 28			Reserved	Reserved
Bit 27				
Bit 26				
Bit 25				
Bit 24				
Bit 23			VBMCLKDIV	
Bit 22				
Bit 21			Reserved	
Bit 20				
Bit 19				
Bit 18				
Bit 17				
Bit 16			VBMBF	CTRSTBF
Bit 15			Reserved	Reserved
Bit 14				
Bit 13				
Bit 12				
Bit 11				
Bit 10				
Bit 9				
Bit 8				
Bit 7				
Bit 6				
Bit 5				
Bit 4	SEGEN		VBMTH	CTRST
Bit 3				
Bit 2				
Bit 1				
Bit 0				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 27.15. LCD0 Memory Map

Register Name	LCD0_SEGDATA4	LCD0_SEGDATA3	LCD0_SEGDATA2	LCD0_SEGDATA1	LCD0_SEGDATA0
ALL Address	0x4004_D0E0	0x4004_D0D0	0x4004_D0C0	0x4004_D0B0	0x4004_D0A0
Access Methods	ALL	ALL	ALL	ALL	ALL
Bit 31	SEGPIN39	SEGPIN31	SEGPIN23	SEGPIN15	SEGPIN7
Bit 30					
Bit 29					
Bit 28					
Bit 27	SEGPIN38	SEGPIN30	SEGPIN22	SEGPIN14	SEGPIN6
Bit 26					
Bit 25					
Bit 24					
Bit 23	SEGPIN37	SEGPIN29	SEGPIN21	SEGPIN13	SEGPIN5
Bit 22					
Bit 21					
Bit 20					
Bit 19	SEGPIN36	SEGPIN28	SEGPIN20	SEGPIN12	SEGPIN4
Bit 18					
Bit 17					
Bit 16					
Bit 15	SEGPIN35	SEGPIN27	SEGPIN19	SEGPIN11	SEGPIN3
Bit 14					
Bit 13					
Bit 12					
Bit 11	SEGPIN34	SEGPIN26	SEGPIN18	SEGPIN10	SEGPIN2
Bit 10					
Bit 9					
Bit 8					
Bit 7	SEGPIN33	SEGPIN25	SEGPIN17	SEGPIN9	SEGPIN1
Bit 6					
Bit 5					
Bit 4					
Bit 3	SEGPIN32	SEGPIN24	SEGPIN16	SEGPIN8	SEGPIN0
Bit 2					
Bit 1					
Bit 0					

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

28. Low Power Oscillator (LPOSC0)

This section describes the Low Power Oscillator (LPOSC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the LPOSC block, which is used by all device families covered in this document.

28.1. Low Power Oscillator Features

The Low Power Oscillator has the following features:

- 20 MHz and divided 2.5 MHz frequencies available for the AHB clock.
- Automatically starts and stops as needed.

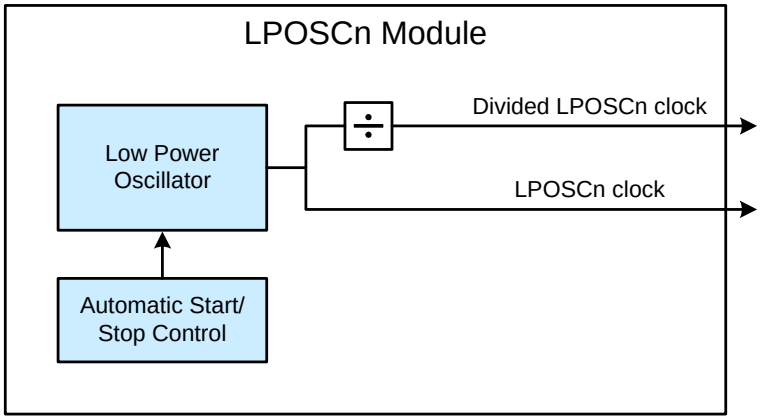


Figure 28.1. Low Power Oscillator Block Diagram

28.2. Operation

The Low Power Oscillator is the default AHB oscillator and enables or disables automatically, as needed. The power consumption of this oscillator is listed in the electrical specifications of the device data sheet. No registers are associated with this peripheral.

The default output frequency of this oscillator is factory calibrated to 20 MHz, and a divided 2.5 MHz version is also available to use as an AHB clock source. More information on the clocks available to the device can be found in the clock control description.

29. Low Power Timer (LPTIMER0)

This section describes the Low Power Timer (LPTIMER) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the LPTIMER block, which is used by all device families covered in this document.

29.1. Low Power Timer Features

The Low Power Timer includes the following features:

- Runs on RTC0TCLK or an external source (rising or falling edge).
- Overflow and compare threshold-match detection, which can generate an interrupt, reset the timer, or wake the device from low power modes.
- Two compare thresholds allow generation of PWM waveforms and periodic pulses.

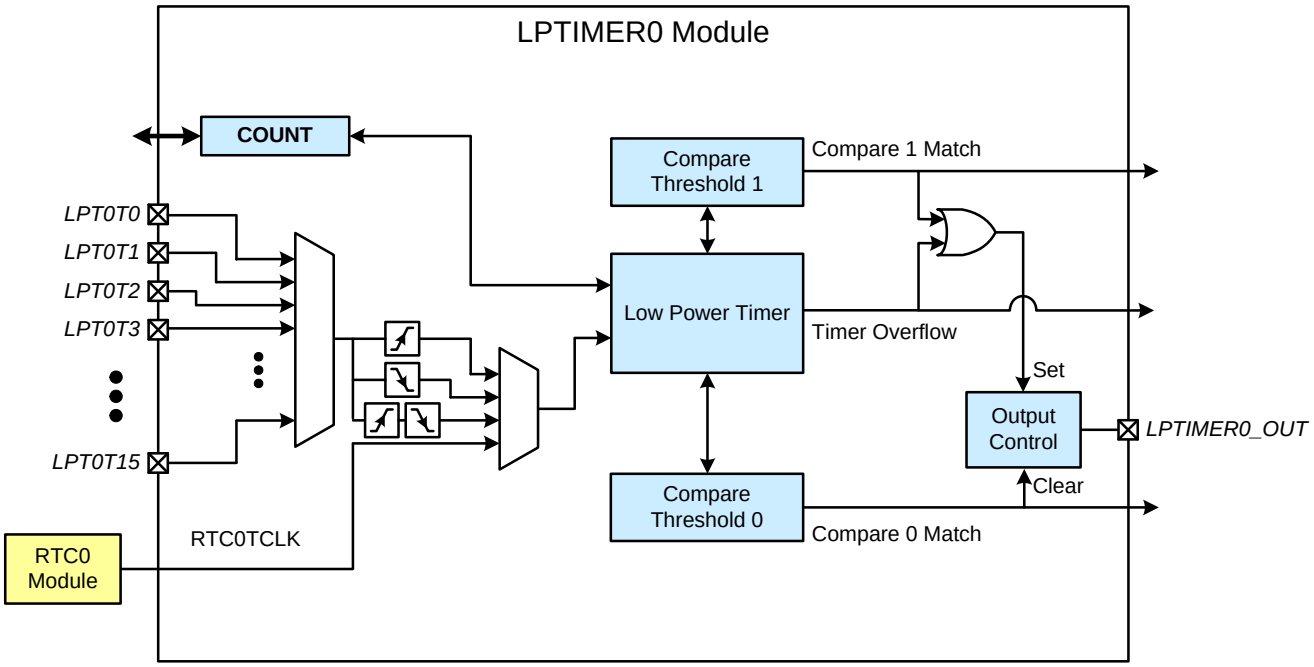


Figure 29.1. Low Power Timer Block Diagram

29.2. Clocking

The Low Power Timer (LPTIMER) module runs from the RTC0 timer clock (RTC0TCLK), allowing the LPTIMER to operate even if the AHB and APB clocks are disabled. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK module clocks to update to a written value.

The LPTIMER counter can increment using one of two clock sources: RTC0TCLK, or rising or falling edges of an external signal.

29.2.1. Clock Input

The LPTIMER timer can use the RTC timer clock (RTC0TCLK) as its input clock. The RTC0TCLK clock is configurable in the RTC0 module as either the RTC0 crystal oscillator (RTC0OSC), an external CMOS clock, or the internal low frequency oscillator (LFOSC0)

29.2.2. External Signal

The LPTIMER module can select one of sixteen external signals as its input clock using the EXTSEL field. The timer can increment on rising, falling, or both edges of the selected external input, controlled by the CMD field. These inputs are synchronized to RTC0TCLK, so they must be high or low for two rising RTC0TCLK edges. Figure 29.2 illustrates the external input signal timing. The external trigger sources available to the LPTIMER vary by package and are defined in Table 29.1.

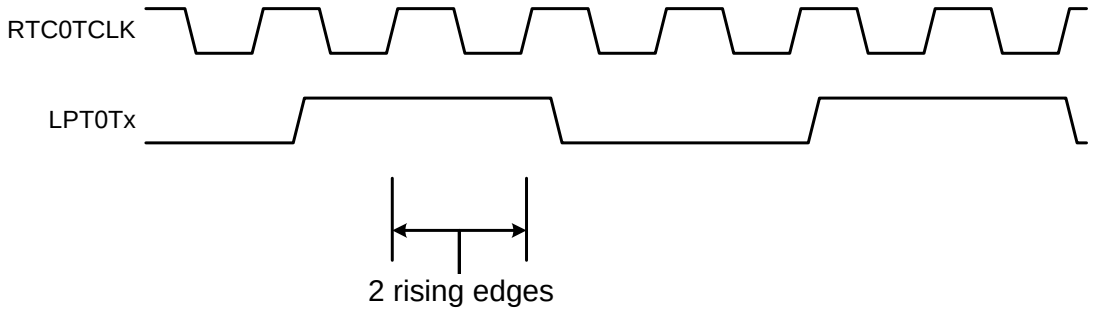


Figure 29.2. External Input Clock Synchronization Timing

Table 29.1. LPTIMER0 Input Mux Selection

LPTIMER0 Trigger	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
LPT0T0	PB0.8	PB0.7	PB0.6
LPT0T1	PB0.9	PB0.8	Reserved
LPT0T2	PB0.10	PB0.9	Reserved
LPT0T3	PB0.11	Reserved	Reserved
LPT0T4	PB1.0	PB1.0	Reserved
LPT0T5	PB1.1	PB1.1	Reserved
LPT0T6	PB1.2	PB1.2	PB0.7

Table 29.1. LPTIMER0 Input Mux Selection

LPTIMER0 Trigger	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
LPT0T7	PB1.3	PB1.3	PB0.8
LPT0T8	PB2.0	PB2.0	PB2.0
LPT0T9	PB2.1	Reserved	PB2.1
LPT0T10	Reserved	Reserved	PB2.2
LPT0T11	Reserved	Reserved	PB2.3
LPT0T12	PB2.4	PB2.4	PB2.4
LPT0T13	PB2.5	PB2.5	PB2.5
LPT0T14	PB2.6	PB2.6	PB2.6
LPT0T15	Comparator 0	Comparator 0	Comparator 0

29.3. Configuring the Timer and Compare Values

The Low Power Timer is an up-counter that has two compare events and an overflow event. Firmware can access the timer value through the TIMER field in the COUNT register and the compare thresholds through the COMPARE0 and COMPARE1 fields in the THRESHOLD register. Note that the RUN bit in the CONTROL register must be set to 1 to allow writes of other bits in the LPTIMER block.

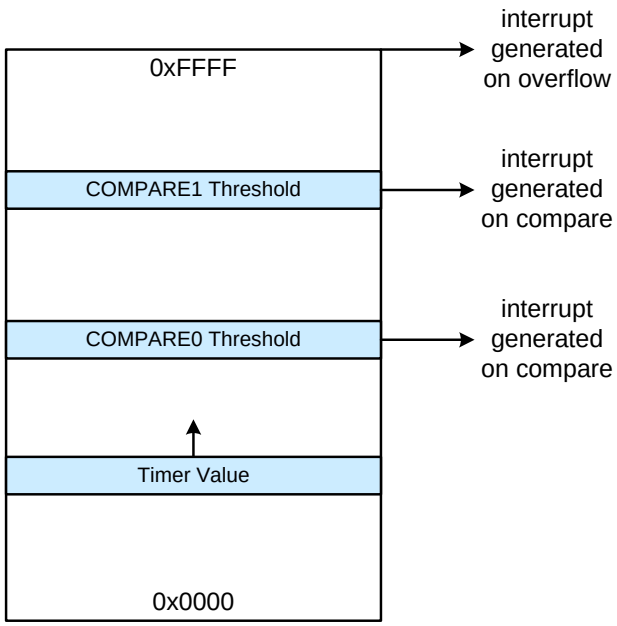


Figure 29.3. Low Power Timer Operation

To write a value to the internal timer register:

1. Write the desired new value to COUNT.
2. Set the TMRSET bit to set the timer threshold.
3. Poll on the TMRSET bit to determine when the operation completes.

To read the current value of the timer:

1. Set the TMRCAP bit to transfer the value from the internal timer to COUNT.
2. Poll on the TMRCAP bit to determine when the operation completes.
3. Read the COUNT register.

If the APB clock is at least four times the speed of the RTC0TCLK, faster access to the COUNT register can be accomplished by setting the HSMDEN bit in the CONTROL register to 1. When this bit is set, the TMRSET and TMRCAP bits do not need to be used, and the COUNT value is accessible via direct writes and reads of the register. When using this mode, the APB bus will be stalled for up to 7 APB clock cycles during the set or capture of the register.

Note: When using a debugger to view the LPTIMER registers, it is recommended to set HSMDEN to 1.

Reading and writing the compare values is accomplished directly through reads and writes of the THRESHOLD register.

Once the LPTIMER timer is configured as desired, firmware can stop and start the timer using the RUN bit. RUN must be set to 1 when changing any register values.

29.4. Interrupts

The LPTIMER module has three sources that can cause an interrupt: timer overflow and comparator 0 or 1 match events. Each can be individually enabled or disabled as needed. The overflow interrupt (OVFI) flag indicates when the timer overflows and can generate an interrupt when OVFIEN is set to 1. Hardware sets the compare 0 interrupt (CMP0I) flag when the timer value matches the value of the COMPARE0 threshold and can generate an interrupt if CMP0IEN is set to 1. Likewise, the compare 1 interrupt (CMP1I) is enabled using the CMP1IEN bit, and triggers off a match of the timer value with the COMPARE1 threshold. The OVFI, CMP0I and CMP1I flags should be cleared by firmware.

29.5. Automatic Reset

Under default operation, the timer will count up from 0x0000 to 0xFFFF and overflow to 0x0000 and repeat the process. It is also possible to automatically reset the timer to zero when either of the comparator match events occur. This automatic reset mode is enabled by setting CMP0RSTEN to 1 for comparator 0 or CMP1RSTEN for comparator 1.

29.6. Output

The LPTIMER module has an output that can optionally be set high on an overflow or comparator 1 event. The output may optionally be set low on a comparator 0 event. This output can be connected to a physical pin using the LPTOSEL bit in the PBCFG0_CONTROL0 register. When the LPTIMER output is enabled, the associated pin should be skipped by the crossbar. The output pin options vary from one package to the next, and are shown in Table 29.2.

Table 29.2. LPTIMER0 Output Selection

LPT0 Output Signal Name	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
LPT0OUT0	PB0.8	PB0.7	PB0.6
LPT0OUT1	PB0.11	PB0.9	Reserved

The OVFOEN, CMP0OEN and CMP1OEN bits in the CONTROL register are used to enable the pin control functions of each event, as shown in Table 29.3.

Table 29.3. Output Pin Events

Event	Bit Value	Effect
Timer overflow	OVFOEN = 0	No effect
	OVFOEN = 1	Output Set to 1
Comparator 0 Match event	CMP0OEN = 0	No effect
	CMP0OEN = 1	Output Cleared to 0
Comparator 1 Match event	CMP1OEN = 0	No effect
	CMP1OEN = 1	Output Set to 1

29.7. Debug Mode

Firmware can set the DBGMD bit to force the LPTIMER module to halt on a debug breakpoint. Clearing the DBGMD bit forces the module to continue operating while the core halts in debug mode.

LPTIMER0 Registers This section contains the detailed register descriptions for LPTIMER0 registers.

Register 29.1. LPTIMER0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RUN	DBGMD	MCLKEN	Reserved			CMP1RSTEN	CMP0RSTEN	Reserved	OUTINVEN	CMP1OEN	CMP1IEN	CMP0OEN	OVFOEN	CMP0IEN	OVFIEN
Type	RW	RW	RW	R			RW	RW	R	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved		OUTEN	CMP1EN	CMP0EN	HSMDEN	TMRCAP	TMRSET	EXTSEL				Reserved		CMD	
Type	R		RW	RW	RW	RW	RW	RW	RW				R		RW	
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LPTIMER0_CONTROL = 0x4003_8000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 29.4. LPTIMER0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	RUN	**Timer Run Control and Compare Threshold Enable.** This bit must be set to '1' to write other bits in the LPTIMER registers. 0: Stop the timer and disable the compare threshold. 1: Start the timer running and enable the compare threshold.
30	DBGMD	**Low Power Timer Debug Mode.** 0: The Low Power Timer module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the Low Power Timer module to halt.
29	MCLKEN	**Low Power Timer Module Clock Enable.** The clock to the LPTIMER0 module must be enabled before accessing the Low Power Timer module registers. Disabling the clock reduces power consumption when the LPTIMER0 module is not in use. 0: Disable the clock to the Low Power Timer module. 1: Enable the clock to the Low Power Timer module.
28:26	Reserved	Must write reset value.

Notes:

1. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK cycles to update to a written value.

Table 29.4. LPTIMER0_CONTROL Register Bit Descriptions

Bit	Name	Function
25	CMP1RSTEN	**Timer Compare 1 Event Reset Enable.** 0: Timer compare 1 events do not reset the timer. 1: Timer compare 1 events reset the timer.
24	CMP0RSTEN	**Timer Compare 0 Event Reset Enable.** 0: Timer compare 0 events do not reset the timer. 1: Timer compare 0 events reset the timer.
23	Reserved	Must write reset value.
22	OUTINVEN	**Output Inversion Enable.** 0: Do not invert the LPTIMER0 output. 1: Invert the LPTIMER0 output.
21	CMP1OEN	**Timer Compare 1 Event Output Enable.** 0: Timer compare 1 events do not modify the Low Power Timer output. 1: Timer compare 1 events set the Low Power Timer output to 1.
20	CMP1IEN	**Timer Compare 1 Event Interrupt Enable.** 0: Disable the timer compare 1 event interrupt. 1: Enable the timer compare 1 event interrupt.
19	CMP0OEN	**Timer Compare 0 Event Output Enable.** 0: Timer compare 0 events do not modify the Low Power Timer output. 1: Timer compare 0 events clear the Low Power Timer output to 0.
18	OVFOEN	**Timer Overflow Output Enable.** 0: Timer overflows do not modify the Low Power Timer output. 1: Timer overflows set the Low Power Timer output to 1.
17	CMP0IEN	**Timer Compare 0 Event Interrupt Enable.** 0: Disable the timer compare 0 event interrupt. 1: Enable the timer compare 0 event interrupt.
16	OVFIEN	**Timer Overflow Interrupt Enable.** 0: Disable the timer overflow interrupt. 1: Enable the timer overflow interrupt.
15:14	Reserved	Must write reset value.
13	OUTEN	**Output Enable.** 0: Disable the LPTIMER0 output. 1: Enable the LPTIMER0 output.
12	CMP1EN	**Timer Compare 1 Threshold Enable.** This bit enables the timer comparator 1 function. When cleared to 0, timer comparator 1 will not generate interrupts or alter the LPT output signal.

Notes:
1. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK cycles to update to a written value.

Table 29.4. LPTIMER0_CONTROL Register Bit Descriptions

Bit	Name	Function
11	CMP0EN	**Timer Compare 0 Threshold Enable.** This bit enables the timer comparator 0 function. When cleared to 0, timer comparator 0 will not generate interrupts or alter the LPT output signal.
10	HSMDEN	**High Speed Timer Access Mode Enable.** When this bit is set to 1, the timer value can be read or written without firmware delays. The APB clock must be at least four times faster than RTC0TCLK.
9	TMRCAP	**Timer Capture.** Writing a 1 to TMRCAP initiates a read of the internal timer register into the COUNT register. This field is automatically cleared by hardware when the operation completes and does not need to be cleared by software.
8	TMRSET	**Timer Set.** Writing a 1 to TMRSET initiates a copy of the value from the COUNT register into the internal timer register. This field is automatically cleared by hardware when the copy is complete and does not need to be cleared by software.
7:4	EXTSEL	**External Trigger Source Select.** 0000: Select external trigger LPTnT0. 0001: Select external trigger LPTnT1. 0010: Select external trigger LPTnT2. 0011: Select external trigger LPTnT3. 0100: Select external trigger LPTnT4. 0101: Select external trigger LPTnT5. 0110: Select external trigger LPTnT6. 0111: Select external trigger LPTnT7. 1000: Select external trigger LPTnT8. 1001: Select external trigger LPTnT9. 1010: Select external trigger LPTnT10. 1011: Select external trigger LPTnT11. 1100: Select external trigger LPTnT12. 1101: Select external trigger LPTnT13. 1110: Select external trigger LPTnT14. 1111: Select external trigger LPTnT15.
3:2	Reserved	Must write reset value.
1:0	CMD	**Count Mode.** 00: The timer is free running mode on the RTC timer clock (RTC0TCLK). 01: The timer is incremented on the rising edges of the selected external trigger (LPTnTx). 10: The timer is incremented on the falling edges of the selected external trigger (LPTnTx). 11: The timer is incremented on both edges of the selected external trigger (LPTnTx).

Notes:
1. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK cycles to update to a written value.

Register 29.2. LPTIMER0_COUNT: Timer Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TIMER															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LPTIMER0_COUNT = 0x4003_8010

Table 29.5. LPTIMER0_COUNT Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	TIMER	**Timer Value.** This field provides read and write access to the internal timer.

Notes:

1. When writing and reading this register, the field may take 3 APB and 2 RTC0TCLK cycles to update to a written value if HSMDEN is cleared to 0.

Register 29.3. LPTIMER0_THRESHOLD: Threshold Values

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	COMPARE1															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	COMPARE0															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LPTIMER0_THRESHOLD = 0x4003_8020

Table 29.6. LPTIMER0_THRESHOLD Register Bit Descriptions

Bit	Name	Function
31:16	COMPARE1	**Timer Compare 1 Threshold Value.** When this field matches the timer value and the compare 1 threshold is enabled (CMP1EN = 1), hardware will set the CMP1I bit to 1. Hardware can also set the LPTIMER0 output to 1 when a match occurs, if enabled (CMP1OEN = 1),
15:0	COMPARE0	**Timer Compare 0 Threshold Value.** When this field matches the timer value and the compare 0 threshold is enabled (CMP0EN = 1), hardware will set the CMP0I bit to 1. Hardware can also clear the LPTIMER0 output to 0 when a match occurs, if enabled (CMP0OEN = 1),

Notes:

1. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK cycles to update to a written value.

Register 29.4. LPTIMER0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved													CMP1I	CMP0I	OVFI
Type	R													RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

LPTIMER0_STATUS = 0x4003_8030

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 29.7. LPTIMER0_STATUS Register Bit Descriptions

Bit	Name	Function
31:3	Reserved	Must write reset value.
2	CMP1I	**Timer Compare 1 Event Interrupt Flag.** Hardware sets this flag to 1 when the timer equals the compare 1 threshold. Firmware must clear this flag and poll until it reads back as zero.
1	CMP0I	**Timer Compare 0 Event Interrupt Flag.** Hardware sets this flag to 1 when the timer equals the compare 0 threshold. Firmware must clear this flag and poll until it reads back as zero.
0	OVFI	**Timer Overflow Interrupt Flag.** Hardware sets this flag to 1 when a timer overflow occurs. Firmware must clear this flag and poll until it reads back as zero.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. When writing and reading registers, a bit or field may take 3 APB and 2 RTC0TCLK cycles to update to a written value.

29.8. LPTIMER0 Register Memory Map

Table 29.8. LPTIMER0 Memory Map

Register Name	LPTIMER0_STATUS	LPTIMER0_THRESHOLD	LPTIMER0_COUNT	LPTIMER0_CONTROL
ALL Address	0x4003_8030	0x4003_8020	0x4003_8010	0x4003_8000
Access Methods	ALL | SET | CLR	ALL	ALL	ALL | SET | CLR
Bit 31	Reserved	COMPARE1	Reserved	RUN
Bit 30				DBGMD
Bit 29				MCLKEN
Bit 28				Reserved
Bit 27				
Bit 26				
Bit 25				CMP1RSTEN
Bit 24				CMP0RSTEN
Bit 23				Reserved
Bit 22				OUTINVEN
Bit 21				CMP1OEN
Bit 20				CMP1IEN
Bit 19				CMP0OEN
Bit 18				OVFOEN
Bit 17				CMP0IEN
Bit 16				OVFIEN
Bit 15		COMPARE0	TIMER	Reserved
Bit 14				
Bit 13				OUTEN
Bit 12				CMP1EN
Bit 11				CMP0EN
Bit 10				HSMDEN
Bit 9				TMRCAP
Bit 8				TMRSET
Bit 7				EXTSEL
Bit 6				
Bit 5				
Bit 4				
Bit 3	CMP1I			Reserved
Bit 2				
Bit 1	CMP0I			CMD
Bit 0	OVFI			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

30. Phase-Locked Loop (PLL0)

This section describes the phase-locked loop (PLL) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the PLL block, which is used by all device families covered in this document.

30.1. PLL Features

The PLL module includes the following features:

- Three output ranges with output frequencies ranging from 23 to 50 MHz (can also be divided by up to 128 to provide lower system clock speeds).
- Multiple reference frequency inputs.
- Three output modes: free-running DCO, frequency-locked, and phase-locked.
- Ability to sense the rising edge or falling edge of the reference source.
- DCO frequency LSB dithering to provide finer average output frequencies.
- Spectrum spreading to reduce generated system noise.
- Low jitter and fast lock times.
- Ability to suspend all output frequency updates (including dithering and spectrum spreading) using the STALL bit during jitter-sensitive measurements.

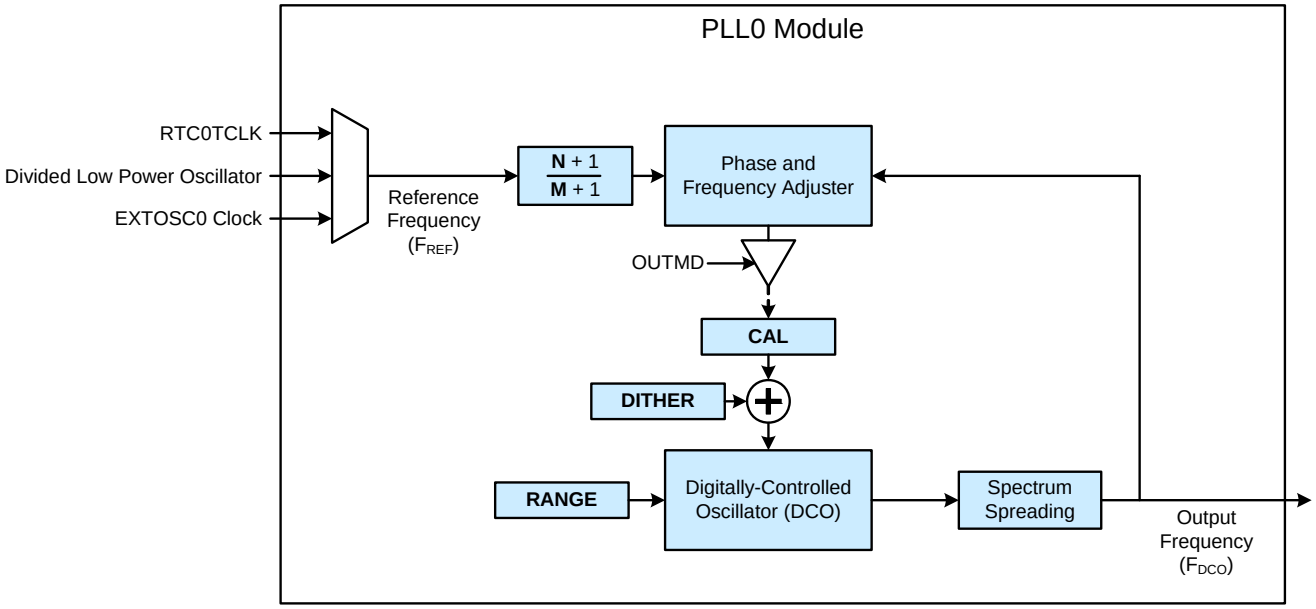


Figure 30.1. PLL Block Diagram

30.2. Overview

The PLL module consists of a dedicated Digitally-Controlled Oscillator (DCO) that can be used in free-running DCO mode without a reference frequency, frequency-locked mode with a reference frequency, or phase-locked mode with a reference frequency. The reference frequency for frequency-lock and phase-lock modes can be sourced from multiple sources (including the RTC timer clock or an external oscillator) to provide maximum flexibility for different application needs. Because the PLL module generates its own clock, the DCO can be locked to a particular reference frequency and then moved to free-running mode to reduce system power or noise.

30.3. Interrupts

A PLL interrupt can be generated whenever the PLL module locks to or unlocks from the reference frequency in either frequency-lock or phase-lock mode. The LCKI flag indicates the locked state of the module, and the LCKSEN bit configures whether the hardware setting (“is locked”) or clearing (“is unlocked”) the LCKI bit causes the interrupt, if enabled (LCKIEN = 1). A PLL interrupt can also be generated whenever the module is trying to lock but saturates (high or low) the DCO period, which indicates that the range should be adjusted. This “is saturated” interrupt is indicated by the HLMT and LLMT flags and is automatically disabled when the module is locked (LCKI = 1). It will automatically become active again if locked status is lost for any reason (LCKI = 0). The LCKI interrupt is level sensitive, and the HLMT and LLMT interrupts are edge sensitive. All three flags (LCKI, HLMT, and LLMT) can be cleared by setting OUTMD to 00b or 01b.

30.4. Output Modes

The PLL module has three available output modes: free-running DCO, frequency-lock, and phase-lock.

30.4.1. Free-Running DCO Mode

The PLL module includes its own Digitally-Controlled Oscillator (DCO) and does not need a reference frequency to generate a clock output. This free-running DCO mode is enabled by setting OUTMD to 01b.

When in free-running DCO mode, the output frequency of the DCO is determined by a combination of the RANGE, CAL, and DITHER settings. The CAL field is factory-calibrated to operate at 49 MHz with RANGE set to 2. These values can be changed to produce other frequencies. Typically this is accomplished in one of two ways:

1. Use Frequency-Lock or Phase-Lock mode to produce the desired output frequency from a reference clock source, and then switch to free-running DCO mode.
2. Program the CAL and RANGE fields directly to pre-determined values. For example, if the PLL has already been used to lock the oscillator to a specific frequency, the value of the CALCONFIG register can be stored for later use in free-running DCO mode.

The spectrum spreading feature is also available in free-running DCO mode. More information on dithering and spectrum spreading can be found in Section 30.5.1 and Section 30.5.2.

30.4.1.1. Free-Running DCO PLL Setup With Factory-Calibrated Settings

To set up the PLL for free-running DCO mode using the factory-calibrated settings:

1. If the output of the DCO will be used as the AHB clock and the AHB clock will be increasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.
2. If it has been modified since a device reset, program the CALCONFIG register to the initial reset value for the device. Note that running the PLL in frequency-lock or phase-lock mode affects the CAL field. Any device reset restores the factory-calibrated value to this register.
3. (Optional) Enable dithering by setting DITHER to a non-zero value.
4. (Optional) Enable spectrum spreading by writing a non-zero value to SSAMP.
5. Set OUTMD to 01b.
6. Switch the AHB clock source to the DCO output using the device’s clock control module.
7. If the output of the DCO will be used as the AHB clock and the AHB clock will be decreasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.

30.4.1.2. Free-Running DCO PLL Setup With Pre-Calibrated Output Frequency

To set up the PLL for free-running DCO mode using a pre-determined calibration value:

1. If the output of the DCO will be used as the AHB clock and the AHB clock will be increasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.
2. Program the CAL field to the maximum value.
3. Program the desired output range by setting the RANGE field.
4. Program the desired output frequency by setting the CAL field.
5. (Optional) Enable dithering by setting DITHER to a non-zero value.
6. (Optional) Enable spectrum spreading by writing a non-zero value to SSAMP.
7. Set OUTMD to 01b.
8. Switch the AHB clock source to the DCO output using the device's clock control module.
9. If the output of the DCO will be used as the AHB clock and the AHB clock will be decreasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.

30.4.1.3. Free-Running DCO PLL Setup Using Frequency-Lock or Phase-Lock Reference

If the device is already operating from the PLL in frequency-lock or phase-lock mode, simply programming OUTMD to 01b will place the PLL in free-running DCO mode. When switching from a different clock source, the recommended procedure is as follows:

1. If the output of the DCO will be used as the AHB clock and the AHB clock will be increasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.
2. Configure the PLL for frequency-lock or phase-lock mode, as detailed in "30.4.5.1. Initial Frequency-Lock and Phase-Lock PLL Setup" to set the desired frequency.
3. (Optional) Enable dithering by setting DITHER to a non-zero value.
4. (Optional) Enable spectrum spreading by writing a non-zero value to SSAMP.
5. Once the PLL is locked, set OUTMD to 01b to switch to free-running DCO mode.
6. Switch the AHB clock source to the DCO output using the device's clock control module.
7. If the output of the DCO will be used as the AHB clock and the AHB clock will be decreasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.

If the CAL, DITHER, or SSAMP DCO settings need to be changed when the output frequency running, these settings can be modified directly even when using the DCO output as the AHB clock. Any changes to RANGE should be made while the DCO output is off (OUTMD = 00b).

30.4.2. Frequency-Lock Mode

In frequency-lock mode, the PLL module will ensure the frequency of the DCO output (F_{DCO}) is derived from the reference frequency (F_{REF}) correctly based on the values of N and M, but the phase of the output frequency may not necessarily match the reference source. Any changes in frequency on the reference will propagate through to the DCO output frequency. frequency-lock mode is enabled by setting OUTMD to 10b.

$$F_{DCO} = F_{REF} \times \frac{N + 1}{M + 1}$$

Equation 30.1. PLL Output Frequency

The hardware will take the firmware-set RANGE, N, and M values and automatically adjust CAL to match the reference frequency. Additionally, finer frequency matching can be achieved by enabling automatic dithering (DITHEN = 1), and generated noise can be reduced by enabling spectrum spreading.

30.4.3. Phase-Lock Mode

Phase-Lock mode matches the phase and frequency of the DCO output (F_{DCO}) to the reference frequency (F_{REF}) based on the values of N and M. Any drift or changes in phase and frequency on the reference will propagate through to the DCO output phase and frequency. Phase-Lock mode is enabled by setting OUTMD to 11b.

$$F_{DCO} = F_{REF} \times \frac{N + 1}{M + 1}$$

The hardware will take the firmware-set RANGE, N, and M values and automatically adjust CAL to match the reference phase and frequency. Additionally, finer frequency matching can be achieved by enabling automatic dithering (DITHEN = 1), and generated noise can be reduced by enabling spectrum spreading.

30.4.4. Frequency-Lock and Phase-Lock Differences

In frequency-lock mode, the frequency error is bounded by definition, but this error will not necessarily approach zero over time. phase-lock mode guarantees the average frequency error approaches zero over time. As a result, the PLL module may need to make DCO output period corrections more often in phase-lock mode, resulting in higher output jitter. Additionally, frequency-lock mode generates less overshoot and undershoot compared to phase-lock mode, resulting in a better transient response. Frequency-lock mode is sufficient for most applications that require a specific output frequency.

30.4.5. Selecting N and M Values

In frequency-lock and phase-lock modes, the output jitter and lock time of the PLL module are dependent upon the N and M factors and the DCO output frequency. Larger values of N and M increase lock time but decrease output jitter. This relationship is shown in Figure 30.2. Longer DCO output periods (slower frequencies) increase lock time.

Equation 30.2 gives the approximate equation for DCO output jitter in frequency-lock or phase-lock modes.

$$\Delta t \sim \pm t_{DCO} \times \left(\frac{1}{N + 1} + \frac{1}{2500} \right)$$

Equation 30.2. Output Jitter

Equation 30.3 and Equation 30.4 approximate the maximum DCO lock time in frequency-lock or phase-lock modes (the PLL may lock faster). In the equations, t_{REF} is the period of the reference clock, t_{DCO} is the period of the DCO output, and t_{INIT} is the period of the DCO before enabling the PLL.

$$t_{lock} \approx t_{REF} \times (M + 1) + 2 \times t_{REF} + 32 \times t_{INIT}$$

Equation 30.3. DCO Lock Time with LOCKTH = 0

$$t_{lock} \approx t_{REF} \times (M + 1) \times (LOCKTH + 1) + 2 \times t_{REF} + 32 \times t_{DCO}$$

Equation 30.4. DCO Lock Time with LOCKTH > 0

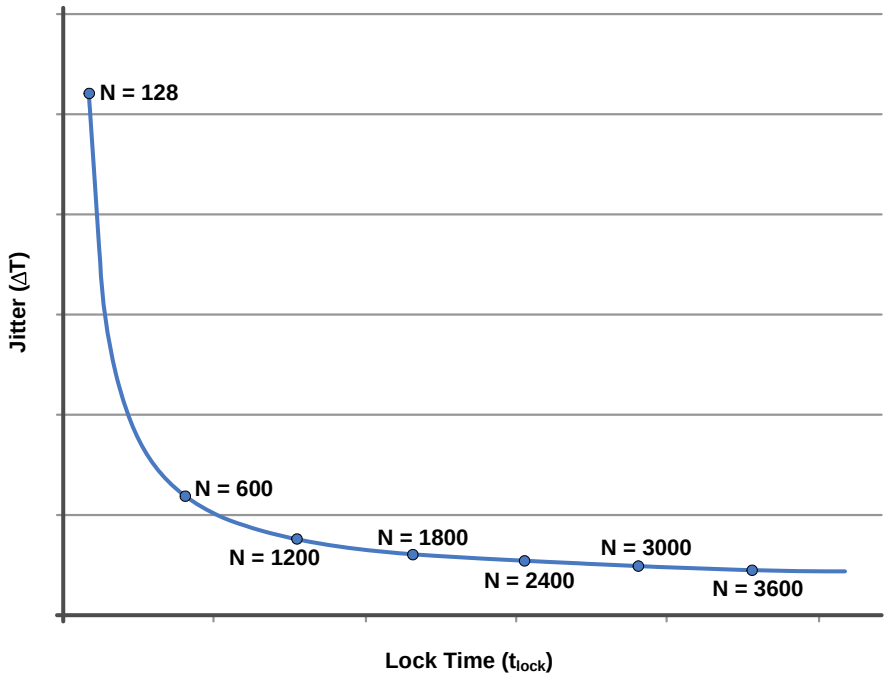


Figure 30.2. Jitter vs. Lock Time for Values of N

The value of M and N should be selected such that the lock time and jitter requirements of the system are met. Alternatively, a small value of N can be used initially to speed the locking process, and then a larger value of N (and consequently M) can be re-written to the registers to reduce output jitter once the module is locked to the reference. Once N is selected, the desired output frequency dictates the corresponding value of M. This method is discussed in detail in Section 30.6.2. The process for updating the PLL register values when the DCO is running is discussed in Section 30.4.5.1.

30.4.5.1. Initial Frequency-Lock and Phase-Lock PLL Setup

To set up the PLL for frequency-lock or phase-lock mode:

1. Ensure that the reference clock to be used (internal or external) is running and stable.
2. If the output of the PLL will be used as the AHB clock and the AHB clock will be increasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.
3. Set the REFSEL bits to select the desired reference clock for the PLL DCO.
4. Program the CAL field to the maximum value.
5. Program the desired output range by setting the RANGE field.
6. Program the appropriate N and M values to achieve the desired output frequency.
7. Select the edge (rising or falling) of the reference frequency the DCO output should lock to using the EDGSEL bit.
8. (Recommended) Enable dithering by setting DITHEN.
9. (Optional) Enable spectrum spreading by writing a non-zero value to SSAMP.
10. (Optional) Enable CAL saturation interrupts by setting LMTIEN to 1.
11. (Optional) Enable PLL interrupts and set LCKSEN to 1 to enable the “is locked” interrupt when the hardware sets the LCKI flag to 1.
12. Set OUTMD to 10b for frequency-lock mode or 11b for phase-lock mode.
13. Wait for the LCKI interrupt (if interrupts are enabled) or poll on LCKI until it changes from 0 to 1. If RANGE is set to the proper value, the HLMT and LLMT saturation interrupts should not occur. If these interrupts are enabled and trigger, this indicates that the RANGE setting is too low or high for the desired output frequency. The RANGE setting can be adjusted by writing OUTMD to 00b, writing RANGE with the new value, and setting OUTMD back to 10b for frequency-lock mode or 11b for phase-lock mode.
14. (Optional) Switch the value of LCKSEN to 0 to cause an interrupt if the PLL module loses lock on the reference frequency.
15. Switch the AHB clock source to the DCO output using the CONTROL register in the CLKCTRL module.
16. If the output of the PLL will be used as the AHB clock and the AHB clock will be decreasing in frequency, program the Flash read timing bits to the appropriate value for the new clock rate.

30.4.5.2. Modifying the RANGE Setting while the PLL is Locked and Running

If the PLL RANGE setting needs to be changed when the output frequency is locked and running, implement the following procedure:

1. Switch the AHB clock to another clock source that is running and stable.
2. Set the OUTMD bits to 00b.
3. Program the CAL field to the maximum value.
4. Modify the RANGE setting as desired.
5. If interrupts are enabled, set LCKSEN to 1 to enable the “is locked” interrupt when the LCKI flag is set to 1.
6. Set the OUTMD bits to 10b for frequency-lock mode or 11b for phase-lock mode. The PLL module will relock to the reference frequency using the new settings.
7. Wait for the LCKI interrupt (if interrupts are enabled) or poll on LCKI until it changes from 0 to 1. If RANGE is set to the proper value, the HLMT and LLMT saturation interrupts should not occur.
8. (Optional) Switch the value of LCKSEN to 0 to cause an interrupt if the PLL module loses lock.

9. Switch the AHB clock source to the DCO output using the CONTROL register in the CLKCTRL module.

30.4.5.3. Modifying PLL Settings (other than RANGE) while the PLL is Locked and Running

If the other PLL settings need to be changed when the output frequency is locked and running, the following procedure should be implemented:

1. Set the OUTMD bits to 01.
2. Modify the settings as desired.
3. If interrupts are enabled, set LCKSEN to 1 to enable the “is locked” interrupt when the LCKI flag is set to 1.
4. Set the OUTMD bits to 10b for frequency-lock mode or 11b for phase-lock mode. The PLL module will relock to the reference frequency using the new settings.
5. Wait for the LCKI interrupt (if interrupts are enabled) or poll on LCKI until it changes from 0 to 1. If RANGE is set to the proper value, the HLMT and LLMT saturation interrupts should not occur.
6. (Optional) Switch the value of LCKSEN to 0 to cause an interrupt if the PLL module loses lock.

If a sensitive analog measurement needs to be taken, the automatic DCO output frequency adjustment (including dithering and spectrum spreading) can be temporarily halted by setting the STALL bit. The module will not resume the output updates until STALL has been cleared to 0 by firmware.

30.5. Additional Features

In addition to three output modes, the PLL module has the additional features of output frequency dithering and spectrum spreading.

30.5.1. Output Frequency Dithering

The CAL field provides 12 bits of frequency steps within a particular output range, determined by RANGE. The output frequency dithering feature allows an average frequency of finer resolution than CAL alone can provide, though the instantaneous frequency will not be equal to the average frequency at a particular moment in time. The dithering setting controls how often a 1 is added to CAL to achieve the desired average frequency. The DITHER value acts like fractional bits of CAL.

In free-running DCO mode (OUTMD set to 01b), firmware can set the dithering to a particular setting by writing the desired value to DITHER. Dithering can be disabled by writing 0’s to DITHER.

In frequency-lock or phase-lock modes (OUTMD set to 10b or 11b), automatic dithering is enabled by setting DITHEN to 1 and will cause the hardware to automatically set these dithering bits to achieve average frequencies the CAL setting would normally not be able to achieve. Disabling automatic dithering causes the hardware to always set DITHER to 0 when in frequency-lock or phase-lock mode.

Setting the STALL bit to 1 will disable dithering until STALL is written with a 0.

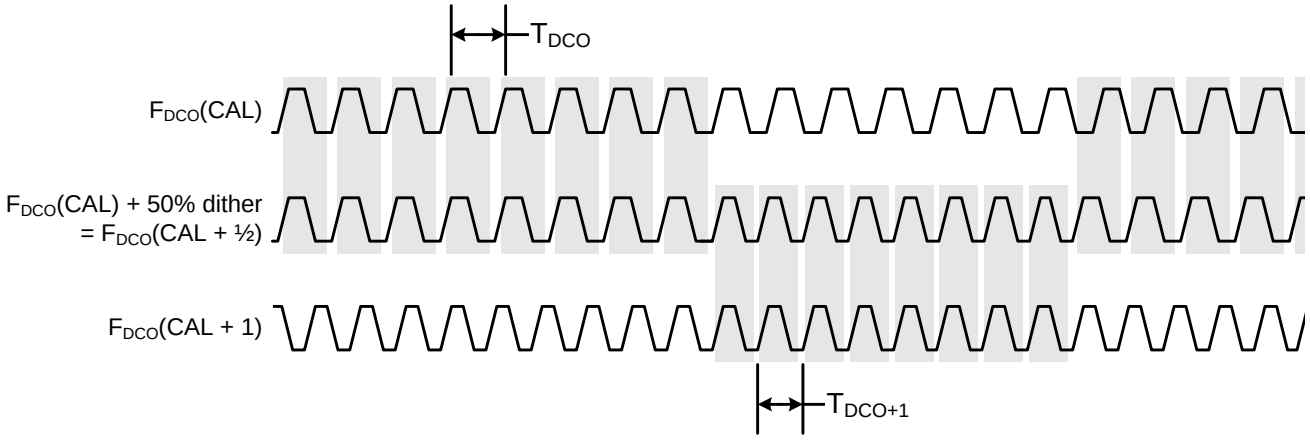


Figure 30.3. Dithering Timing Example

30.5.2. Output Frequency Spectrum Spreading

Spectrum spreading adds intentional noise to the DCO output period to improve system noise performance. Spectrum spreading is a signed random number of uniform distribution added to the DCO output period to slightly vary the resulting clock output. This random number has an adjustable amplitude based on the SSAMP field and an adjustable update rate based on the SSUINV field. Setting the STALL bit to 1 will disable spectrum spreading until STALL is written with a 0.

Note: The added noise from spectrum spreading is centered about the DCO output period. As a result, the peak frequencies may exceed the maximum allowable frequency for the AHB clock. As a result, the desired output frequency created by RANGE, N, and M should be programmed below the maximum allowable frequency to allow some headroom for spectrum spreading.

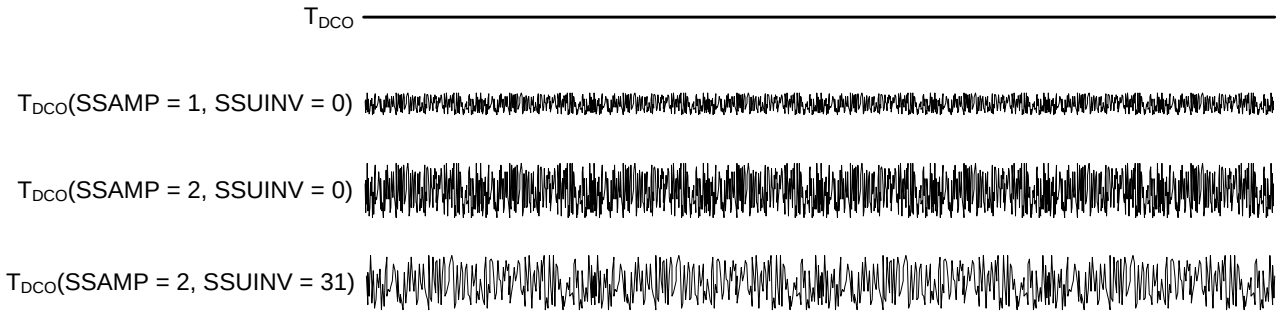


Figure 30.4. Spectrum Spreading Example

The spectrum spreading amplitude has fixed settings of approximately ± 0.1%, ± 0.2%, ± 0.4%, ± 0.8%, or ± 1.6% of the DCO output period. Equation 30.5 describes the spectrum spreading update interval.

$$\text{UpdateInterval} = 4 \times T_{DCO} \times (SSUINV + 1)$$

Equation 30.5. Spectrum Spreading Update Rate

30.6. Advanced Setup Examples

This section discusses advanced uses of the PLL module to save power or reduce the external components required by a system.

30.6.1. Temporarily Using a Reference

The PLL module can temporarily use a reference to tune the DCO to a frequency. Once the DCO is locked to the reference, the PLL can be switched to free-running DCO mode and the reference source can be disabled. This enables the system to operate at the desired frequency without providing power to the reference source. The DCO output created in this manner will remain stable at the correct frequency, but may not have tolerances as tight as the original reference source. The reference can periodically be enabled and the DCO allowed to relock to improve tolerances, if desired.

To use the PLL module in this manner, follow the procedure in Section 30.4.5.1 for frequency-lock mode and then set OUTMD to 01b for free-running DCO mode. The hardware-tuned CAL and DITHER settings will remain until overwritten by either firmware or hardware (OUTMD set to 10b or 11b). To ensure good performance, it is recommended that LOCKTH be set to a non-zero value when locking to a temporary reference.

30.6.2. Obtaining Fast Lock Times and Low Output Jitter

As discussed in Section 30.4.4, the value of N affects both the lock time and the output jitter of the DCO output frequency. To obtain both a fast lock time and low output jitter, a small value of N can be used initially to speed the locking process, and then a larger value of N can be re-written to the registers to reduce output jitter once the module is locked to the reference.

For example, a DCO output frequency of approximately 50 MHz is desired using a reference clock of 20 MHz. Using Equation 30.1, if (N+1) is 130, the corresponding value of (M+1) is 52 for a desired output frequency of 50 MHz. If (N+1) is 3000, the corresponding value of (M+1) is 1200.

To obtain a fast lock time and a long-term low output jitter:

1. Follow the procedure in Section 30.4.5.1 for frequency-lock mode, setting N to 129 and M to 51.
2. Once the DCO output is LCKI, set OUTMD to 01b for free-running DCO mode.
3. Change N to 2999 and M to 1199.
4. Set OUTMD back to 10b for frequency-lock mode or 11b for phase-lock mode.

30.7. PLL0 Registers

This section contains the detailed register descriptions for PLL0 registers.

Register 30.1. PLL0_DIVIDER: Reference Divider Setting

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved				N											
Type	R				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				M											
Type	R				RW											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PLL0_DIVIDER = 0x4003_B000

Table 30.1. PLL0_DIVIDER Register Bit Descriptions

Bit	Name	Function
31:28	Reserved	Must write reset value.
27:16	N	**N Divider Value.** Selects the reference clock multiplier. Valid values include 32 to 4095 (values below 31 are invalid and may cause incorrect behavior). The frequency-lock or phase-lock output frequency is the result of the equation: $F_{DCO} = F_{REF} \times \frac{N+1}{M+1}$
15:12	Reserved	Must write reset value.

Notes:

1. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Table 30.1. PLL0_DIVIDER Register Bit Descriptions

Bit	Name	Function
11:0	M	**M Divider Value.** Selects the reference clock divider. Valid values include the entire 12-bit range of M (0 to 4095). The frequency-lock or phase-lock output frequency is the result of the equation: $F_{DCO} = F_{REF} \times \frac{N + 1}{M + 1}$

Notes:

1. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Register 30.2. PLL0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OUTMD		EDGSEL	DITHEN	Reserved	STALL	Reserved				LOCKTH		Reserved		REFSEL	
Type	RW		RW	RW	RW	RW	R				RW		R		RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved				LCKPOL	LCKIEN	LMTIEN	Reserved						LCKI	HLMTF	LLMTF
Type	R				RW	RW	RW	R						R	R	R
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PLL0_CONTROL = 0x4003_B010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 30.2. PLL0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:30	OUTMD	**PLL Output Mode.** Sets the DCO output mode. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode. 00: DCO output is off. 01: DCO output is in Free-Running DCO mode. 10: DCO output is in frequency-lock mode (reference source required). 11: DCO output is in phase-lock mode (reference source required).

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Table 30.2. PLL0_CONTROL Register Bit Descriptions

Bit	Name	Function
29	EDGSEL	**Edge Lock Select.** Selects between locking on the rising edge or falling edge of the reference frequency in frequency-lock and phase-lock modes. The setting of this bit has no effect in Free-Running DCO mode. 0: Lock DCO output frequency to the falling edge of the reference frequency. 1: Lock DCO output frequency to the rising edge of the reference frequency.
28	DITHEN	**Dithering Enable.** Enables automatic dithering on the DCO output period in frequency-lock and phase-lock modes. The DITHER field is used to generate a 1-bit dither of the DCO output frequency. This provides better performance and generally reduces the overall jitter in frequency-lock and phase-lock modes. Automatic dithering can be disabled by clearing DITHEN to 0, which forces the hardware to always generate DITHER values of 0. When OUTMD is set to 01b, firmware can write a non-zero value to the DITHER bits to enable dithering, even if DITHEN is 0. Setting DITHEN to 1 in Free-Running DCO mode will have no effect. 0: Automatic DCO output dithering disabled. 1: Automatic DCO output dithering enabled.
27	Reserved	Must write reset value.
26	STALL	**DCO Output Updates Stall.** When STALL is set to 1, the phase-lock and frequency-lock modes are temporarily disabled. Spectrum spreading and dithering are also disabled. This is useful for providing the lowest jitter environment possible for sensitive analog measurements. The phase-lock and frequency-lock modes, spectrum spreading, and dithering, are re-enabled when STALL is cleared to 0. 0: In phase-lock and frequency-lock modes, spectrum spreading, and dithering operate normally, if enabled. 1: In phase-lock and frequency-lock modes, spectrum spreading, and dithering are prevented from updating the output of the DCO.
25:22	Reserved	Must write reset value.
21:20	LOCKTH	**Lock Threshold Control.** Lock is deterministic based on the number of (M+1)xTREF cycles over which the algorithm has operated. Assuming lock is possible given the present RANGE setting, phase lock is guaranteed after one (M+1)xTREF cycle. The value of LOCKTH sets the number of extra (M+1)xTREF cycles to wait before declaring lock. It is recommended to set the LOCKTH field to 1 if the PLL will be switched to free-running mode immediately after lock.
19:18	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Table 30.2. PLL0_CONTROL Register Bit Descriptions

Bit	Name	Function
17:16	REFSEL	**Reference Clock Selection Control.** 00: PLL reference clock (FREF) is the RTC0 oscillator (RTC0TCLK). 01: PLL reference clock (FREF) is the divided Low Power Oscillator (LPOSC0). 10: PLL reference clock (FREF) is the external oscillator output (EXTOSC0). 11: Reserved.
15:12	Reserved	Must write reset value.
11	LCKPOL	**Lock Interrupt Polarity.** Sets the state of LCKI that causes the PLL lock state interrupt to occur, if enabled. 0: The lock state PLL interrupt will occur when LCKI is 0. 1: The lock state PLL interrupt will occur when LCKI is 1.
10	LCKIEN	**Locked Interrupt Enable.** 0: The PLL locking does not cause an interrupt 1: An interrupt is generated if LCKI matches the state selected by LCKPOL.
9	LMTIEN	**Limit Interrupt Enable.** Enables interrupt generation if the DCO output frequency saturates high or low, preventing the DCO from reliably locking to the reference frequency or phase. This interrupt will not be generated while the DCO is locked (LCKI = 1). 0: Saturation (high and low) interrupt disabled. 1: Saturation (high and low) interrupt enabled.
8:3	Reserved	Must write reset value.
2	LCKI	**Phase-Lock and Frequency-Lock Locked Interrupt Flag.** Indicates when the PLL module DCO is locked to the reference clock. In phase-lock mode, this indicates that the DCO is phase-locked with the reference clock. In frequency-lock mode, this indicates that the DCO is frequency-locked (not necessarily phase-locked) with the reference clock. This bit will also assert the level-sensitive lock state interrupt, if enabled. This bit is automatically set and cleared by hardware, but can also be manually cleared by writing 00b or 01b to OUTMD. 0: DCO is disabled or not locked. 1: DCO is enabled and locked.
1	HLMTF	**CAL Saturation (High) Flag.** Indicates when the DCO output period is saturated high and the DCO cannot lock reliably, so the RANGE value should be decreased. This flag is not automatically cleared by hardware and must be cleared by software by writing 00b or 01b to OUTMD. 0: DCO period is not saturated high. 1: DCO period is saturated high.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Table 30.2. PLL0_CONTROL Register Bit Descriptions

Bit	Name	Function
0	LLMTF	**CAL Saturation (Low) Flag.** Indicates when the DCO output period is saturated low and the DCO cannot lock reliably, so the RANGE value should be increased. This flag is not automatically cleared by hardware and must be cleared by software by writing 00b or 01b to OUTMD. 0: DCO period is not saturated low. 1: DCO period is saturated low.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Register 30.3. PLL0_SSPR: Spectrum Spreading Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			SSUINV					Reserved					SSAMP		
Type	R			RW					R					RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
PLL0_SSPR = 0x4003_B020

Table 30.3. PLL0_SSPR Register Bit Descriptions

Bit	Name	Function
31:13	Reserved	Must write reset value.
12:8	SSUINV	**Spectrum Spreading Update Interval.** The update interval is given by the following equation: $UpdateInterval = 4 \times T_{DCO} \times (SSUINV + 1)$
7:3	Reserved	Must write reset value.
2:0	SSAMP	**Spectrum Spreading Amplitude.** 000: Disable Spectrum Spreading. 001: Spectrum Spreading set to approximately ± 0.1% of TDCO. 010: Spectrum Spreading set to approximately ± 0.2% of TDCO. 011: Spectrum Spreading set to approximately ± 0.4% of TDCO. 100: Spectrum Spreading set to approximately ± 0.8% of TDCO. 101: Spectrum Spreading set to approximately ± 1.6% of TDCO. 110-111: Reserved.

Notes:
1. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

Register 30.4. PLL0_CALCONFIG: Calibration Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved													RANGE		
Type	R													RW		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CAL												DITHER			
Type	RW												RW			
Reset	X	X	X	X	X	X	X	X	X	X	X	X	0	0	0	0

Register ALL Access Address

PLL0_CALCONFIG = 0x4003_B030

Table 30.4. PLL0_CALCONFIG Register Bit Descriptions

Bit	Name	Function
31:19	Reserved	Must write reset value.
18:16	RANGE	**DCO Range.** Determines the output frequency range of the DCO. 000: DCO operates from 23 to 37 MHz. 001: DCO operates from 33 to 50 MHz. 010: DCO operates from 45 to 50 MHz. 011-111: Reserved.
15:4	CAL	**DCO Calibration Value.** This value adjusts the output period of the PLL module DCO in fine steps. Increasing CAL increases the DCO output period and decreases the DCO output frequency.
3:0	DITHER	**DCO Dither Setting.** The DITHER bits control how often a 1 is added to CAL to create an average frequency in between the two CAL settings (CAL and CAL + 1). The DITHER value acts like fractional bits of CAL. In Free-Running DCO mode, firmware can write these bits to select a specific dithering setting. Writing 0's to this field disables dithering. The value of DITHEN has no effect in this mode. When DITHEN is set to 1 in frequency-lock or phase-lock modes, the hardware will automatically adjust the DITHER bits. If DITHEN is set to 0 in these modes, the hardware will force the DITHER bits to 0.

Notes:

1. All PLL register fields except LCKPOL, LMTIEN, and STALL must remain static while OUTMD is set to frequency-lock or phase-lock modes. If RANGE setting changes are required, turn off the DCO output, write the maximum value to CAL, write to RANGE, and re-enable frequency-lock or phase-lock mode. All other settings can be changed by putting the module in Free-Running Mode, writing to the registers, and re-enabling frequency-lock or phase-lock mode.

30.8. PLL0 Register Memory Map

Table 30.5. PLL0 Memory Map

Register Name	PLL0_CALCONFIG	PLL0_SSPR	PLL0_CONTROL	PLL0_DIVIDER	Bit
ALL Address	0x4003_B030	0x4003_B020	0x4003_B010	0x4003_B000	
Access Methods	ALL	ALL	ALL | SET | CLR	ALL	
			OUTMD	Reserved	31
			EDGSEL		30
			DITHEN		29
			Reserved		28
			STALL		27
			Reserved		26
			LOCKTH		25
			Reserved		24
			REFSEL		23
			Reserved		22
			LCKPOL		21
			LCKIEN		20
			LMTIEN		19
			Reserved		18
			LCKI		17
			HLMTF		16
			LLMTF		15
		Reserved		Reserved	14
					13
					12
		SSUINV			11
					10
					9
		Reserved			8
					7
					6
		SSAMP			5
					4
					3
	Reserved			N	2
					1
					0
	RANGE			Reserved	15
					14
					13
					12
	CAL			M	11
					10
					9
					8
					7
					6
					5
					4
					3
	DITHER				2
					1
					0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

31. Process/Voltage/Temperature Oscillator (PVTOSC0)

This section describes the Process Voltage/Temperature (PVTOSC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the PVTOSC block, which is used by all device families covered in this document.

31.1. PVTOSC Features

The Process/Voltage/Temperature monitor consists of two modules (TIMER2 and PVTOSC0) designed to monitor the digital circuit performance of the SiM3L1xx device.

The PVT oscillator (PVTOSC0) consists of two oscillators, one operating from the memory LDO and the other operating from the digital LDO. These oscillators have two independent speed options and provide the clocks for two 16-bit timers in the TIMER2 module using the EX input. By monitoring the resulting counts of the TIMER2 timers, firmware can monitor the current device performance and adjust the scalable LDO regulator output voltages as needed to save power.

The PVTOSC module includes the following features:

- Two separate oscillators and timers for the memory and digital logic voltage domains.
- Two oscillator output divider settings.
- Coupled with TIMER2 to provide a method for monitoring digital performance to allow firmware to adjust the scalable LDO regulator output voltages to the lowest level possible, saving power.

31.2. PVTOSC Operation

The optimal PVTOSC settings for different operational speeds are to be determined (TBD), pending full characterization of silicon over process, temperature, and voltage corners. More information on operating the PVTOSC will be available when characterization is complete.

31.3. PVTOSC0 Registers

This section contains the detailed register descriptions for PVTOSC0 registers.

Register 31.1. PVTOSC0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									CLKSEL	MEMOSCMD	DIGOSCMD	Reserved		MEMOSCEN	DIGOSCEN
Type	R									RW	RW	RW	R	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

PVTOSC0_CONTROL = 0x4003_D000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 31.1. PVTOSC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:7	Reserved	Must write reset value.
6	CLKSEL	**Clock Select.** 0: Select the digital and memory oscillators as the inputs to the clock dividers. 1: Select the APB clock as the input to the clock dividers.
5	MEMOSCMD	**High Voltage Oscillator Mode.** 0: Select fast mode for the memory LDO PVT oscillator. 1: Select slow mode for the memory LDO PVT oscillator.
4	DIGOSCMD	**Digital LDO Oscillator Mode.** 0: Select fast mode for the digital LDO PVT oscillator. 1: Select slow mode for the digital LDO PVT oscillator.
3:2	Reserved	Must write reset value.
1	MEMOSCEN	**Memory LDO Oscillator Enable.** 0: Disable the memory LDO PVT oscillator. 1: Enable the memory LDO PVT oscillator.
0	DIGOSCEN	**Digital LDO Oscillator Enable.** 0: Disable the digital LDO PVT oscillator. 1: Enable the digital LDO PVT oscillator.

31.4. PVTOSC0 Register Memory Map

Table 31.2. PVTOSC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
PVTOSC0_CONTROL	0x4003_D000	ALL | SET | CLR	Reserved																									CLKSEL	MEMOSCMD	DIGOSCMD	Reserved		MEMOSCEN	DIGOSCEN

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

32. Real Time Clock and Low Frequency Oscillator (RTC0)

This section describes the Real Time Clock and Low Frequency Oscillator (RTC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the RTC block, which is used by all device families covered in this document.

32.1. RTC Features

The RTC module includes the following features:

- 32-bit timer (supports up to 36 hours) with three separate alarms.
- Option for one alarm to automatically reset the RTC timer.
- Missing clock detector.
- Module clock may be sourced from the internal low frequency oscillator, an external 32.768 kHz crystal, or an external CMOS clock.
- Programmable internal loading capacitors support a wide range of external 32.768 kHz crystals. No additional resistors or capacitors necessary.
- Operates directly from VBAT and remains operational even when the device goes into its lowest power down mode.
- Buffered RTC timer clock output (RTC0TCLK_OUT) provides an accurate, low frequency clock to external devices.

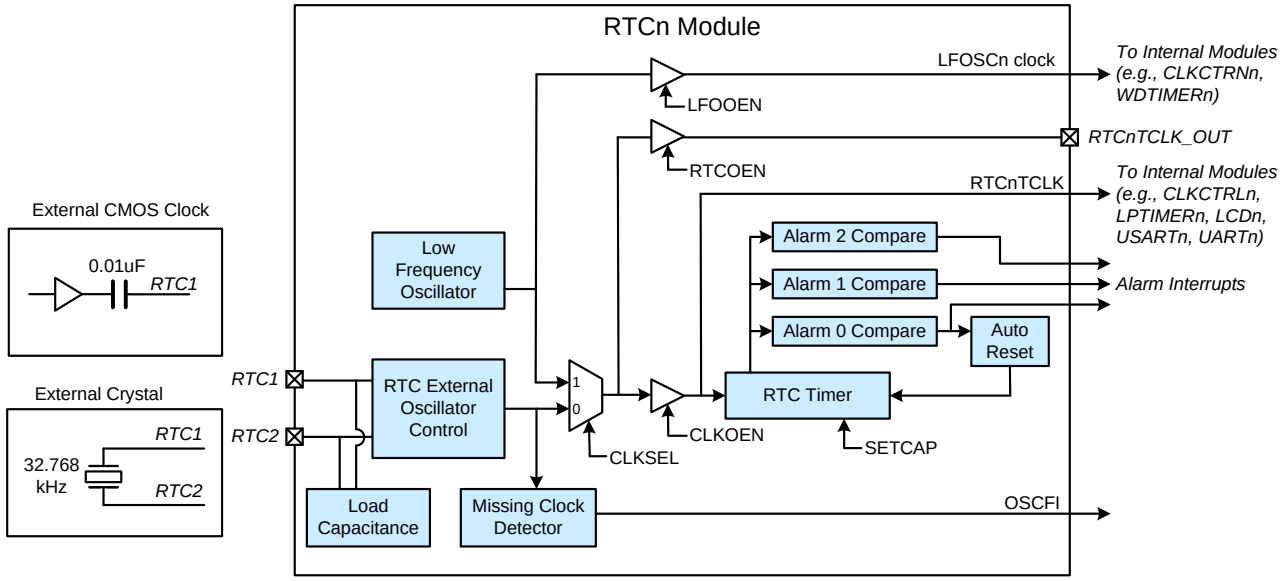


Figure 32.1. RTC Block Diagram

32.2. Overview

The RTC module allows a maximum of 36 hour 32-bit independent time-keeping when used with a 32.768 kHz watch crystal. The RTC provides three alarm events in addition to a missing clock event, which can also function as interrupt, reset or wakeup sources.

The RTC module includes internal loading capacitors that are programmable to 16 discrete levels, allowing compatibility with a wide range of crystals.

The RTC timer clock (RTC0TCLK) can be buffered and routed to a port bank pin to provide an accurate, low frequency clock to other devices while the core is in its lowest power down mode. The module also includes a low power internal low frequency oscillator that reduces sleep mode current and is available for other modules to use as a clock source.

32.3. Clocking

The RTC module clocks from its own timebase independent of the AHB clock. This timebase can be derived from an external 32.768 kHz crystal, an external CMOS clock, or an internal low frequency oscillator.

32.3.1. Crystal Mode

When using the external crystal mode, a 32.768 kHz crystal should be connected between RTC1 and RTC2. No other external components are required. The following steps show how to start the RTC crystal oscillator in firmware:

1. Disable automatic gain control (AGCEN = 0) and enable bias doubling (BDEN = 1).
2. Enable automatic load capacitance stepping (ASEN = 1) or program the RTCLC field directly.
3. Select the RTC oscillator (CLKSEL = 0).
4. Enable the crystal oscillator (CRYSEN = 1).
5. Wait 20 ms.
6. Poll the clock valid (CLKVF) until the crystal oscillator stabilizes.
7. Poll the load capacitance ready (LRDYF) flag until the load capacitance reaches its programmed value.
8. Enable automatic gain control (AGCEN = 1) and disable bias doubling (BDEN = 0) for maximum power savings.
9. Enable the missing clock detector (MCLKEN = 1).
10. Wait 2 ms.
11. Clear OSCFI.
12. If using the RTC Timer, enable the RTC Timer (RTCEN = 1) and the RTC Clock Output Enable (CLKOEN = 1).
13. If serving as a wake up source from a low-power mode, clear the wake-up source flags in the device power management module.

Figure 32.2 shows the hardware configuration for crystal mode.

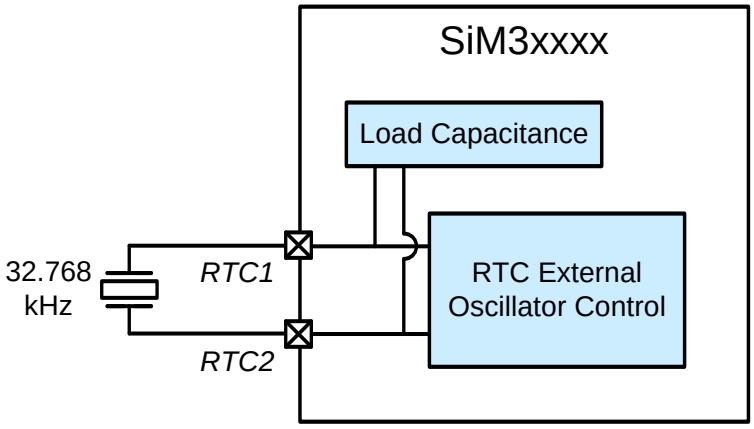


Figure 32.2. Crystal Mode Hardware Configuration

32.3.2. External CMOS Clock Mode

The RTC oscillator may also be driven by an external CMOS clock. The CMOS clock should be applied to RTC1 through a 0.01 µF ac-coupling capacitor, and RTC2 should be left floating. In this mode, the external CMOS clock should have a minimum voltage swing of 400 mV without exceeding VBAT or dropping below VSS.

To use the module with an external CMOS clock:

1. Disable automatic gain control (AGCEN = 0) and disable bias doubling (BDEN = 0).
2. Select the lowest bias setting (RTCLC = 0).
3. Select the RTC oscillator (CLKSEL = 0).
4. Enable the crystal oscillator (CRYSEN = 1).
5. Wait 2 ms.
6. Enable the missing clock detector (MCLKEN = 1).
7. Wait 2 ms.
8. Check the OSCFI flag to ensure a valid clock is present on RTC1.
9. If using the RTC Timer, enable the RTC Timer (RTCEN = 1) and the RTC Clock Output Enable (CLKOEN = 1).
10. If serving as a wake up source from a low-power mode, clear the wake-up source flags in the device power management module.

The CLKVF bit is indeterminate when using a CMOS clock with the RTC module.

Figure 32.3 shows the hardware configuration for external CMOS clock mode.

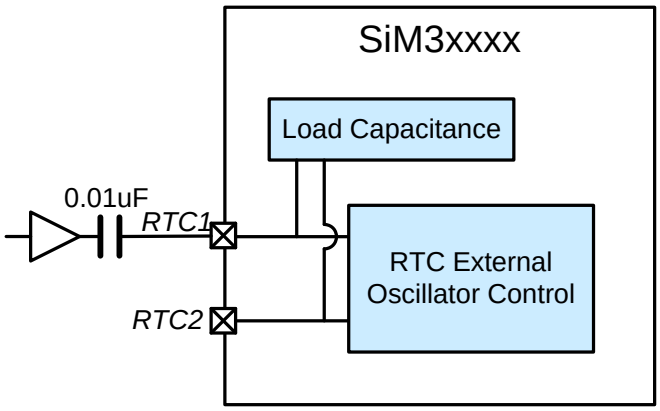


Figure 32.3. External CMOS Clock Mode Hardware Configuration

32.3.3. Low Frequency Oscillator

The low frequency oscillator (LFOSC0) provides a low power internal clock source for the RTC timer. No external components are required to use the low frequency oscillator and the RTC1 and RTC2 pins do not need to be shorted together. The typical oscillation frequency of the low frequency oscillator is 16.4 kHz, but may vary depending on supply voltage, temperature and process. Consult the electrical characteristics tables in the device data sheet for more information.

To use the low frequency oscillator with the RTC module:

1. Enable the low frequency oscillator (LFOSCEN = 1). The oscillator starts oscillating instantaneously.
2. Select the low frequency oscillator as the RTC timer clock source (CLKSEL = 1).
3. Disable the crystal oscillator (CRYSEN = 0).
4. If using the RTC Timer, enable the RTC Timer (RTCEN = 1) and the RTC Clock Output Enable (CLKOEN

= 1).

5. If serving as a wake up source from a low-power mode, clear the wake-up source flags in the device power management module.

When using the low frequency oscillator as its clock source, the RTC module increments bit 1 of the 32-bit timer instead of bit 0, effectively multiplying the oscillator frequency by 2.

32.3.4. RTC Timer Clock Selection

The RTC timer clock (RTC0TCLK) is configured by the RTC Timer Clock Select (CLKSEL) bits to be either the RTC Crystal Oscillator(RTC0OSC), external CMOS clock, or the low frequency oscillator (LFOSC0).

32.3.4.1. RTC Timer Clock Output to Internal Modules

If the RTC clock output is enabled (CLKOEN = 1), the RTC timer clock (RTC0TCLK) is available for use as a clock source by the RTC timer and other internal modules.

32.3.4.2. RTC Timer Clock Output to External Pin

If the RTC external output is enabled (RTCOEN = 1), the RTC timer clock (RTC0TCLK) is output to an external IO pin. Consult data sheet pin definition for location of the RTC0TCLK_OUT function.

32.3.4.3. LFO Output to Internal Modules

If the Low Frequency Oscillator Output is enabled (LFOOEN = 1), the LFOSC0 clock is output to the WDTIMERn and CLKCTRLn modules. To save power, this bit can be disabled before entering PM8.

32.3.5. Programmable Load Capacitance

The programmable load capacitance has 16 values to support a wide range of crystal oscillators. If automatic load capacitance stepping is enabled (ASEN = 1), the crystal load capacitors start at the smallest setting to allow a fast startup time, then slowly increase the capacitance until reaching the final programmed value in the RTCLC field. The RTCLC setting specifies the amount of internal load capacitance and does not include any stray PCB capacitance. Once the final programmed loading capacitance value is reached, the hardware will set the load ready (LRDYF) flag to 1.

Table 32.1 shows the equivalent crystal load capacitance for RTCLC settings.

Table 32.1. Load Capacitance Settings

RTCLC Value	Crystal Load Capacitance	Equivalent Capacitance seen on RTC1 and RTC2
0	4.0 pF	8.0 pF
1	4.5 pF	9.0 pF
2	5.0 pF	10.0 pF
3	5.5 pF	11.0 pF
4	6.0 pF	12.0 pF
5	6.5 pF	13.0 pF
6	7.0 pF	14.0 pF
7	7.5 pF	15.0 pF
8	8.0 pF	16.0 pF
9	8.5 pF	17.0 pF
10	9.0 pF	18.0 pF

Table 32.1. Load Capacitance Settings

RTCLC Value	Crystal Load Capacitance	Equivalent Capacitance seen on RTC1 and RTC2
11	9.5 pF	19.0 pF
12	10.5 pF	21.0 pF
13	11.5 pF	23.0 pF
14	12.5 pF	25.0 pF
15	13.5 pF	27.0 pF

32.3.6. Automatic Gain Control (Crystal Mode Only) and Bias Doubling

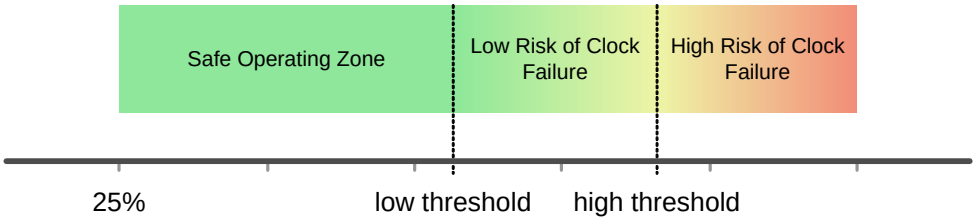
Automatic gain control (AGC) allows the RTC oscillator to trim the oscillation amplitude of a crystal in order to achieve the lowest possible power consumption. Automatic gain control automatically detects when the oscillation amplitude has reached a point where it safe to reduce the drive current, and it may be enabled during crystal startup. It is recommended to enable AGC in most systems that use the RTC oscillator in crystal mode. The following are recommended crystal specifications and operating conditions when enabling AGC:

- ESR < 50 kΩ
- Load capacitance < 10 pF
- Supply voltage < 3.0 V
- Temperature > –20 °C

The chosen crystal should undergo an oscillation robustness test to ensure it will oscillate under the worst case condition to which the system will be exposed. This worst case condition will occur at the following system conditions: lowest temperature, highest supply voltage, highest ESR, highest load capacitance, and lowest bias current (AGC enabled, bias doubling disabled).

To perform the oscillation robustness test, the RTC oscillator output should be routed to a port pin configured as a push-pull digital output using the device port configuration module. The positive duty cycle of the output clock can be used as an indicator of oscillation robustness.

As shown in Figure 32.4, duty cycles less than the low threshold indicate a robust oscillation. As the duty cycle approaches the high threshold, oscillation becomes less reliable and the risk of clock failure increases. Increasing the bias current by disabling AGC will always improve oscillation robustness and will reduce the output clock's duty cycle. This test should be performed at the worst case system conditions, as results at very low temperatures or high supply voltage will vary from results taken at room temperature or low supply voltage. Consult the device data sheet for information on the robust duty cycle range specifications.



RTCn Oscillator (RTCnOSC)
Duty Cycle

Figure 32.4. Interpreting Oscillation Robustness (Duty Cycle) Test Results

AGC may be disabled at the cost of increased power consumption. Disabling Automatic Gain Control will provide the crystal oscillator with higher immunity against the external factors that may cause clock failure.

The bias doubling feature can increase the self-oscillation frequency and allow a higher crystal drive strength in crystal mode. High crystal drive strength is recommended when the crystal is exposed to poor environmental conditions, including excessive moisture. The bias doubler should always be enabled during oscillator startup. Note that when the bias doubler is disabled, the RTC External Oscillator Valid Flag (CLKVF) is disabled and will always read 0.

Table 32.2 shows a summary of the oscillator AGC and bias doubling settings.

Table 32.2. RTC Bias Settings

Bias Doubling (BDEN) Setting	AGC (AGCEN) Setting	Power Consumption
off (0)	on (1)	lowest
off (0)	off (0)	low
on (1)	on (1)	high
on (1)	off (0)	highest

32.3.7. Missing Clock Detector

The missing clock detector (MCD) is a one-shot circuit enabled by setting MCLKEN to 1. When the MCD is enabled, hardware sets the oscillator fail (OSCFI) flag if the RTC oscillator (RTC0OSC) frequency drops below the missing clock detector trigger frequency given in the device data sheet.

The missing clock detector can only be used for the external crystal oscillator or external CMOS clock modes, and should be disabled if using the low frequency oscillator clock as the RTC time clock.

An MCD timeout can trigger an interrupt, wake the device from a low power mode, or reset the device. This feature should be disabled when making changes to the oscillator settings to avoid undesired interrupts or resets.

32.3.8. Oscillator Crystal Valid Detector

The RTC oscillator crystal valid detector is an oscillation amplitude detector circuit used during crystal startup to determine when oscillation is nearly stable. Firmware can read the output of this detector using the clock valid (CLKVF) flag in the CONTROL register. The output of CLKVF is not valid for the first 2 ms after turning on the crystal oscillator. The CLKVF bit is always low when bias doubling is disabled (BDEN = 0).Therefore, the bias doubler should be enabled during crystal startup.

The crystal valid detector is not intended for detecting an oscillator failure - once set, the CLKVF will not be reset unless the external oscillator is disabled. To determine if the oscillator has failed, firmware should use the missing clock detector and OSCFI flag.

32.4. Accessing the Timer

The RTC timer is a 32-bit counter that increments every RTC oscillator cycle. Firmware can set the value of the timer by writing a 32-bit value to the SETCAP register and setting the TMRSET. Hardware will automatically clear TMRSET when the set operation completes. Firmware can read the current value of the timer by setting the TMRCAP bit. Hardware will automatically clear TMRCAP when the capture operation completes, and firmware can then read the SETCAP register.

If the AHB clock is greater than or equal to 4X the RTC clock, the RTC High Speed Mode Enable bit (HSMDEN) must be set to allow the timer value to be accessed.

If the AHB clock is less than 4X the RTC clock, the HSMDEN bit should be cleared. To set the timer when HSMDEN = 0, the timer must be running (RUN = 1).

32.5. Alarms

The RTC timer has three alarm functions that can be set to generate an interrupt, wake the device from a low power mode, reset the device at a specific time, or reset the timer count.

The alarms can be set using the ALARM0, ALARM1, and ALARM2 registers. These 32-bit fields are compared directly to the 32-bit timer value. The alarms are enabled using the ALM0EN, ALM1EN, and ALM2EN bits. Hardware sets the ALM0I, ALM1I, and ALM2I when the corresponding ALARMx value matches the timer, generating an interrupt, if enabled.

32.5.1. Automatic Timer Reset

The RTC timer includes an automatic reset feature that resets the timer to zero when alarm 0 triggers.

When using this auto-reset feature, the alarm match value should always be set to 2 counts less than the desired match value to account for delays. When using the low frequency oscillator in combination with auto-reset, the right-justified alarm 0 value should be set to 4 counts less than the desired match value.

The auto-reset feature can be enabled by writing a 1 to ALM0AREN when Alarm 0 is enabled (ALM0EN = 1).

32.6. Interrupts

The RTC module has interrupts for each of the alarms and the missing clock detector. The ALM0I, ALM1I, and ALM2I flags can cause an interrupt if the corresponding alarm is enabled (ALM0EN, ALM1EN, or ALM2EN set to 1). Hardware sets the oscillator fail (OSCFI) flag to 1 when a missing clock detector event triggers, causing an interrupt.

Firmware can only use the SET and CLR addresses when modifying interrupt flags to avoid hardware conflicts.

32.7. Usage Models

The RTC timer and alarms have two operating modes to suit varying applications.

32.7.1. Usage Mode 1

The first mode uses the RTC timer as a perpetual timebase that is never reset to zero. Every 36 hours, the timer is allowed to overflow without being stopped or disrupted. Firmware manages the alarm intervals and adds the intervals to the expired value in the ALARMx registers after each alarm. This allows the alarm match value to always stay ahead of the timer by one firmware-managed interval. If firmware uses 32-bit unsigned addition to increment the alarm match values, then it does not need to handle overflows since both the timer and the alarm match values will overflow in the same manner.

This mode is ideal for applications using a long alarm interval (24 or 36 hours) or have a need for a perpetual timebase, which is useful in situations where the wake-up interval is constantly changing. For these applications, firmware can keep track of the number of timer overflows in a 16-bit variable, extending the 32-bit (36 hour) timer to a 48-bit (272 year) perpetual timebase.

32.7.2. Usage Mode 2

The second mode uses the RTC timer as a general purpose up-counter that is auto-reset to zero by hardware after each alarm 0 event. Hardware manages the alarm intervals in the ALARMx registers, and firmware only needs to set the alarm intervals once during device initialization. After each alarm 0 event, firmware should keep a count of the number of alarms that have occurred in order to keep track of time. Alarm 1 and alarm 2 events do not trigger the timer auto-reset.

This mode is ideal for applications that require minimal firmware intervention or have a fixed alarm interval. This mode is the most power-efficient since it requires less core processing time per alarm. Alarm 0 can be configured to reset the counter after 24 hours, thus keeping a daily schedule for the other alarms.

32.8. RTC0 Registers

This section contains the detailed register descriptions for RTC0 registers.

Register 32.1. RTC0_CONFIG: RTC Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RTCEN	CLKSEL	RTCOEN	CLKOEN	Reserved	ALM2EN	ALM1EN	ALM0EN	Reserved					AGCEN	CRYSEN	BDEN
Type	RW	RW	RW	RW	R	RW	RW	RW	R					RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								RTCLC				ASEN	MCLKEN	RUN	ALM0AREN
Type	R								RW				RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

RTC0_CONFIG = 0x4002_9000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 32.3. RTC0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	RTCEN	**RTC Timer Enable.** 0: Disable the RTC timer. 1: Enable the RTC timer.
30	CLKSEL	**RTC Timer Clock Select.** 0: Select the External Crystal or External CMOS Clock as the RTC Timer clock (RTCnTCLK) source. 1: Select the Low Frequency Oscillator as the RTC Timer clock (RTCnTCLK) source.
29	RTCOEN	**RTC External Output Enable.** Setting this bit to 1 allows the RTC module to drive the RTCnTCLK on the external RTCnTCLK_OUT pin. When the output is enabled on a pint, the associated pin should be skipped in the crossbar. 0: Disable the external RTCnTCLK_OUT output. 1: Enable the external RTCnTCLK_OUT output.
28	CLKOEN	**RTC Clock Output Enable.** 0: Disable the RTCnTCLK output to the timer and other internal modules. 1: Enable the RTCnTCLK output to the timer and other internal modules.

Table 32.3. RTC0_CONFIG Register Bit Descriptions

Bit	Name	Function
27	Reserved	Must write reset value.
26	ALM2EN	**Alarm 2 Enable.** 0: Disable RTC Alarm 2. 1: Enable RTC Alarm 2.
25	ALM1EN	**Alarm 1 Enable.** 0: Disable RTC Alarm 1. 1: Enable RTC Alarm 1.
24	ALM0EN	**Alarm 0 Enable.** 0: Disable RTC Alarm 0. 1: Enable RTC Alarm 0.
23:19	Reserved	Must write reset value.
18	AGCEN	**Automatic Gain Control Enable.** 0: Disable automatic gain control. 1: Enable automatic gain control, saving power.
17	CRYSEN	**Crystal Oscillator Enable.** 0: Disable the crystal oscillator circuitry. 1: Enable the crystal oscillator circuitry.
16	BDEN	**Bias Doubler Enable.** 0: Disable the bias doubler, saving power. 1: Enable the bias doubler.
15:8	Reserved	Must write reset value.
7:4	RTCLC	**Load Capacitance Value.** This field is the load capacitance value. This field will be automatically set by hardware if automatic load capacitance stepping is enabled (ASEN = 1).
3	ASEN	**Automatic Crystal Load Capacitance Stepping Enable.** 0: Disable automatic load capacitance stepping. 1: Enable automatic load capacitance stepping.
2	MCLKEN	**Missing Clock Detector Enable.** 0: Disable the missing clock detector. 1: Enable the missing clock detector. If the missing clock detector triggers, it will generate an RTC Fail event.
1	RUN	**RTC Timer Run Control.** 0: Stop the RTC timer. 1: Run the RTC timer.
0	ALM0AREN	**Alarm 0 Automatic Reset Enable.** Setting this bit to 1 will automatically reset the RTC timer when a Alarm 0 event occurs. Note that Alarm 0 must be enabled (ALM0EN=1) to use the Automatic Reset feature. 0: Disable the Alarm 0 automatic reset. 1: Enable the Alarm 0 automatic reset.

Register 32.2. RTC0_CONTROL: RTC Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved							LRDYF	HSMDEN	OSCFI	CLKVF	TMRSET	TMRCAP	ALM2I	ALM1I	ALM0I
Type	R							R	RW	RW	R	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	X	0	0	0	0	0	0	0	0

Register ALL Access Address

RTC0_CONTROL = 0x4002_9010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 32.4. RTC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:9	Reserved	Must write reset value.
8	LRDYF	**RTC Load Capacitance Ready Flag.** This bit is set by hardware when the load capacitance matches the programmed value. 0: The load capacitance is currently stepping. 1: The load capacitance has reached its programmed value.
7	HSMDEN	**RTC High Speed Mode Enable.** This bit should be set to 1 by firmware if the AHB clock is greater than or equal to 4x the RTC Timer Clock (RTCnTCLK) frequency. 0: Disable high speed mode. (AHBCLK < 4x RTCnTCLK) 1: Enable high speed mode. (AHBCLK >= 4x RTCnTCLK)
6	OSCFI	**RTC Oscillator Fail Interrupt Flag.** This bit is set by hardware when a missing clock detector timeout occurs. This bit must be cleared by software. 0: Oscillator is running. 1: Oscillator has failed.
5	CLKVF	**RTC External Oscillator Valid Flag.** 0: External oscillator is not valid. 1: External oscillator is valid.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 32.4. RTC0_CONTROL Register Bit Descriptions

Bit	Name	Function
4	TMRSET	**RTC Timer Set.** Set this bit to 1 to initiate an RTC timer set operation, which copies the value in SETCAP to the RTC timer. If HSMDEN=0, the timer must be running (RUN=1) in order to set the timer value. This bit is cleared by hardware when the transfer operation is complete. 0: RTC timer set operation is complete. 1: Start the RTC timer set.
3	TMRCAP	**RTC Timer Capture.** Set this bit to 1 to initiate an RTC timer capture operation, which copies the current RTC timer value to SETCAP. This bit is cleared by hardware when the transfer operation is done. 0: RTC timer capture operation is complete. 1: Start the RTC timer capture.
2	ALM2I	**Alarm 2 Interrupt Flag.** 0: Alarm 2 event has not occurred. 1: Alarm 2 event occurred.
1	ALM1I	**Alarm 1 Interrupt Flag.** 0: Alarm 1 event has not occurred. 1: Alarm 1 event occurred.
0	ALM0I	**Alarm 0 Interrupt Flag.** 0: Alarm 0 event has not occurred. 1: Alarm 0 event occurred.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 32.3. RTC0_ALARM0: RTC Alarm 0

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALARM0[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ALARM0[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
RTC0_ALARM0 = 0x4002_9020

Table 32.5. RTC0_ALARM0 Register Bit Descriptions

Bit	Name	Function
31:0	ALARM0	**RTC Alarm 0.** The RTC Alarm 0 event will occur when ALARM0 matches the RTC timer value.

Register 32.4. RTC0_ALARM1: RTC Alarm 1

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALARM1[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ALARM1[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
RTC0_ALARM1 = 0x4002_9030

Table 32.6. RTC0_ALARM1 Register Bit Descriptions

Bit	Name	Function
31:0	ALARM1	**RTC Alarm 1.** The RTC Alarm 1 event will occur when ALARM1 matches the RTC timer value.

Register 32.5. RTC0_ALARM2: RTC Alarm 2

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ALARM2[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ALARM2[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
RTC0_ALARM2 = 0x4002_9040

Table 32.7. RTC0_ALARM2 Register Bit Descriptions

Bit	Name	Function
31:0	ALARM2	**RTC Alarm 2.** The RTC Alarm 2 event will occur when ALARM2 matches the RTC timer value.

Register 32.6. RTC0_SETCAP: RTC Timer Set/Capture Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SETCAP[31:16]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	SETCAP[15:0]															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

RTC0_SETCAP = 0x4002_9050

Table 32.8. RTC0_SETCAP Register Bit Descriptions

Bit	Name	Function
31:0	SETCAP	**RTC Timer Set/Capture Value.** The value in SETCAP will be written to the RTC timer when TMRSET is set to 1. The operation will be complete when TMRSET is cleared to 0 by the hardware. The value of the RTC timer will be copied to SETCAP when TMRCAP is set to 1. The operation will be complete when TMRCAP is cleared to 0 by the hardware.

Register 32.7. RTC0_LFOCONTROL: LFOSC Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	LFOSCEN	LFOOEN	Reserved													
Type	RW	RW	R													
Reset	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

RTC0_LFOCONTROL = 0x4002_9060

Table 32.9. RTC0_LFOCONTROL Register Bit Descriptions

Bit	Name	Function
31	LFOSCEN	**Low Frequency Oscillator Enable.** 0: Disable the Low Frequency Oscillator (LFOSCn). 1: Enable the Low Frequency Oscillator (LFOSCn).
30	LFOOEN	**Low Frequency Oscillator Output Enable.** 0: Disable the Low Frequency Oscillator output to internal modules. 1: Enable the Low Frequency Oscillator output to internal module.
29:0	Reserved	Must write reset value.

32.9. RTC0 Register Memory Map

Table 32.10. RTC0 Memory Map

Register Name	RTC0_CONFIG	RTC0_CONTROL	RTC0_ALARM0	RTC0_ALARM1	RTC0_ALARM2
ALL Address	0x4002_9000	0x4002_9010	0x4002_9020	0x4002_9030	0x4002_9040
Access Methods	ALL | SET | CLR	ALL | SET | CLR	ALL	ALL	ALL
Bit 31	RTCEN	Reserved	ALARM0	ALARM1	ALARM2
Bit 30	CLKSEL				
Bit 29	RTCOEN				
Bit 28	CLKOEN				
Bit 27	Reserved				
Bit 26	ALM2EN				
Bit 25	ALM1EN				
Bit 24	ALM0EN				
Bit 23	Reserved				
Bit 22					
Bit 21					
Bit 20					
Bit 19					
Bit 18	AGCEN				
Bit 17	CRYSEN				
Bit 16	BDEN				
Bit 15	Reserved				
Bit 14					
Bit 13					
Bit 12					
Bit 11					
Bit 10					
Bit 9					
Bit 8		LRDYF			
Bit 7	RTCLC	HSMDEN			
Bit 6		OSCFI			
Bit 5		CLKVF			
Bit 4		TMRSET			
Bit 3	ASEN	TMRCAP			
Bit 2	MCLKEN	ALM2I			
Bit 1	RUN	ALM1I			
Bit 0	ALM0AREN	ALM0I			

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 32.10. RTC0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RTC0_SETCAP	0x4002_9050	ALL	SETCAP																															
RTC0_LFOCONTROL	0x4002_9060	ALL	LFOSCEN	LFOOEN	Reserved																													

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

33. SAR Analog-to-Digital Converter (SARADC0)

This section describes the SAR Analog to Digital Converter (SARADC) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the SARADC block, which is used by SARADC0 on all device families covered in this document.

33.1. SARADC Features

The SARADC module includes the following features:

- Single-ended 10-bit or 12-bit operation.
- Operation in low power modes at lower conversion speeds.
- Can be synchronized to the EPCA0 synchronization output, to take samples at precise times in the PWM waveform.
- Selectable asynchronous hardware conversion trigger with hardware channel select.
- Output data window comparator allows automatic range checking.
- Support for Burst Mode, which produces one set of accumulated data per conversion-start trigger with programmable power-on settling and tracking time.
- Conversion complete, multiple conversion complete, and FIFO overflow and underflow flags and interrupts supported.
- Flexible output data formatting.
- Sequencer allows up to 8 sources to be automatically scanned using one of four channel characteristic profiles without software intervention.
- Eight-word conversion data FIFO for DMA operations.

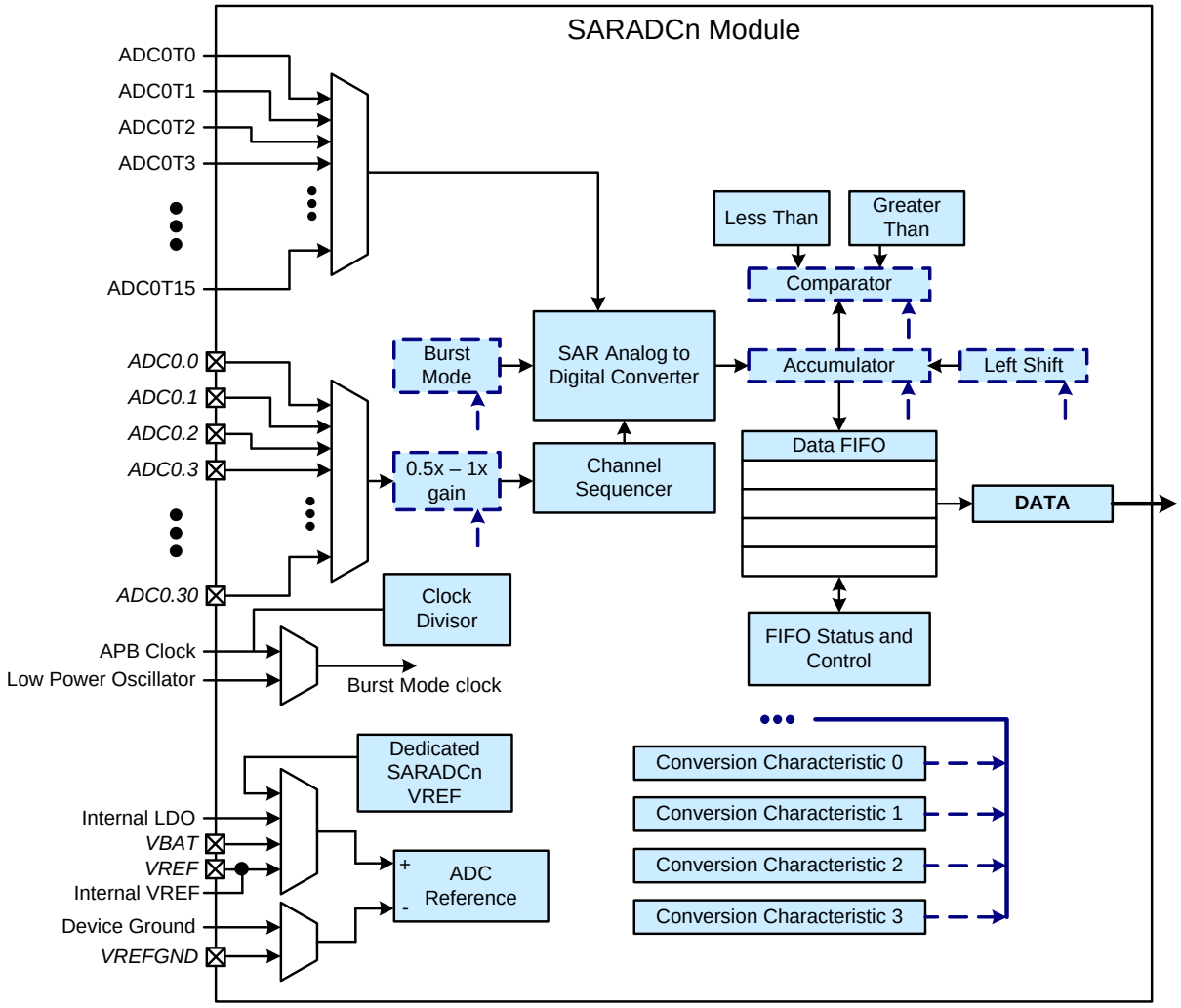


Figure 33.1. SARADC Block Diagram

33.2. Tracking and Conversion Time

A single ADC conversion consists of two phases; the tracking phase, and the conversion phase. During the tracking phase, the ADC's sampling capacitor connects to the selected multiplexer channel and charges to the voltage present at that node. During the conversion phase, the sampling capacitor disconnects from the multiplexer channel and connects to the SAR converter circuitry.

33.2.1. Input Settling Time

It is important for the application to allow enough settling time at the ADC input during the tracking phase. This will depend largely on the external source impedance and the desired accuracy level. The input to the SARADC is shown in Figure 33.2. The sample switch is closed during the tracking phase and open during the conversion phase. Values for C_{IN}, C_{SAR}, and R_{MUX} are different depending on the type of input channel and the gain range. These values can be found in the device data sheet electrical specifications tables. The system designer should assume that the capacitor C_{SAR} may have an arbitrary starting charge on every conversion.

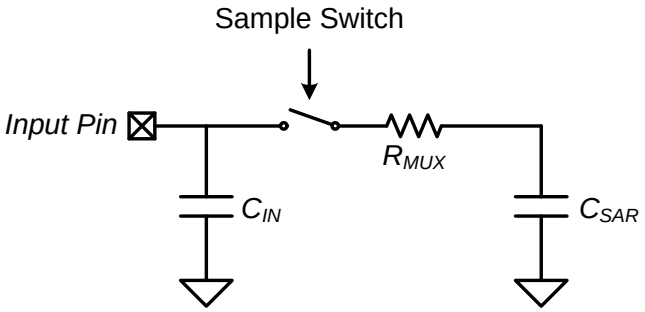


Figure 33.2. SARADC Input Model

33.2.2. SAR Clock Generation

The SAR clock speed dictates the timing of the conversion phase, which lasts for 13 SAR clock cycles. The SAR clock speed is programmable as a divided version of the APB clock using the CLKDIV field in the CONFIG register. The SAR clock should be configured to be as fast as possible according to the electrical specifications in the device data sheet. Faster SAR clock speeds allow the converter more time in the tracking phase and reduce the amount of time the converter is actively converting, thereby reducing system power.

33.2.3. Tracking Mode (Non-Burst Operation)

When the ADC operates in non-burst mode, two tracking options are available and are selected by the TRKMD bit in the CONTROL register: normal tracking and delayed tracking. In normal tracking mode, the ADC tracks any time a conversion is not taking place, and the start-of-conversion event triggers the beginning of the conversion phase. The ADC returns to tracking immediately after a conversion finishes. The tracking time in this mode is therefore determined by the sample interval minus the conversion time.

In delayed tracking mode, the start-of-conversion event will trigger the ADC to track for three SAR clock cycles, followed by the conversion phase. Upon completion of a conversion, the ADC will go into an idle state, waiting for the next start-of-conversion trigger. Timing for the two tracking modes is shown in Figure 33.3.

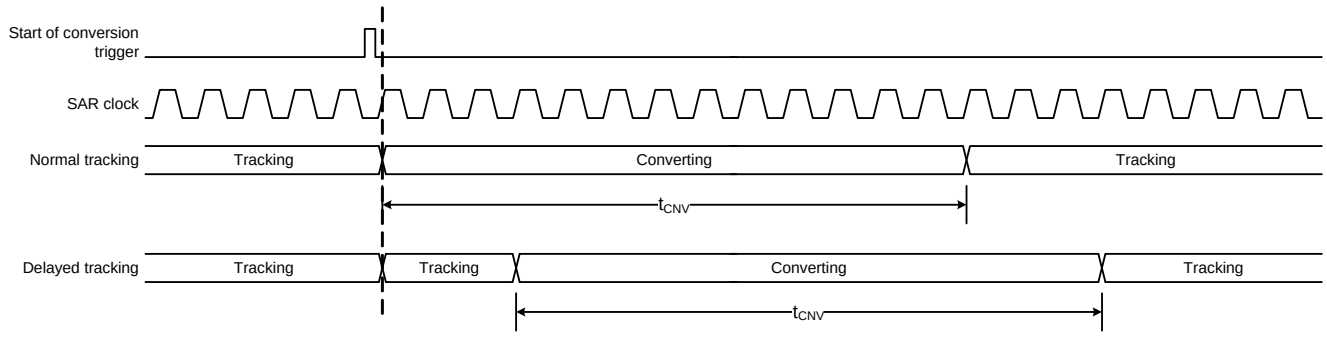


Figure 33.3. Non-Burst Tracking and Conversion Timing

33.2.4. Start-of-Conversion Source

Conversions can be initiated by several different internal and external trigger sources. The SCSEL field in the CONTROL register selects the start-of-conversion source to be used by the ADC. In "on-demand" trigger mode, conversions are initiated when firmware sets the ADBUSY bit in the CONTROL register to 1. In all other conversion modes, the selected start-of-conversion trigger (timer overflow, external pin transition, etc.) will begin a conversion. The frequency of the selected start-of-conversion trigger determines the sampling rate of the ADC and should not exceed the maximum listed in the electrical specification for the device. SARADC0 start of conversion sources vary by package, and are shown in Table 33.1.

Table 33.1. SARADC0 Start of Conversion Sources

Trigger	External Convert Start Description	SiM3U1x7/C1x7 Pin Name
ADC0T0	Internal Convert Start	“On Demand” by writing 1 to ADBUSY
ADC0T1	Internal Convert Start	Timer 0 Low overflow
ADC0T2	Internal Convert Start	Timer 0 High overflow
ADC0T3	Internal Convert Start	Timer 1 Low overflow
ADC0T4	Internal Convert Start	Timer 1 High overflow
ADC0T5	Internal Convert Start	EPCA0 synchronization pulse
ADC0T6	Internal Convert Start	I2C0 Timer overflow
ADC0T15	External Convert Start	ADC0T15 routed through crossbar

33.2.5. 12-bit Mode

The ADC normally operates as a 10-bit converter, and achieves its highest sampling rate in the 10-bit mode. A 12-bit mode is available, which increases the resolution of the converter to 12 bits at the expense of conversion speed.

This 12-bit mode is a special condition of burst mode operation. When operating the converter in 12-bit mode, the BURSTEN bit in the CONTROL register must be set to 1. One input sample and four conversion cycles are required per 12-bit conversion word, and the set of four conversions will be initiated by the start-of-conversion trigger. The converter uses a patented technique to increase the resolution and linearity of the converter by two bits using only four conversion cycles, whereas a traditional straight average operation would require 16 samples to increase noise resolution by 2 bits, and would have no effect on linearity. Additionally, when used to sample DC input signals, the converter can be configured to sample the input four times per 12-bit conversion, which can provide additional filtering of Gaussian noise present at the input. The AD12BSSEL field on the CONTROL register is used to configure whether four separate input samples or a single input sample is used for the 12-bit result. Figure 33.4 shows the difference in timing between these two options.

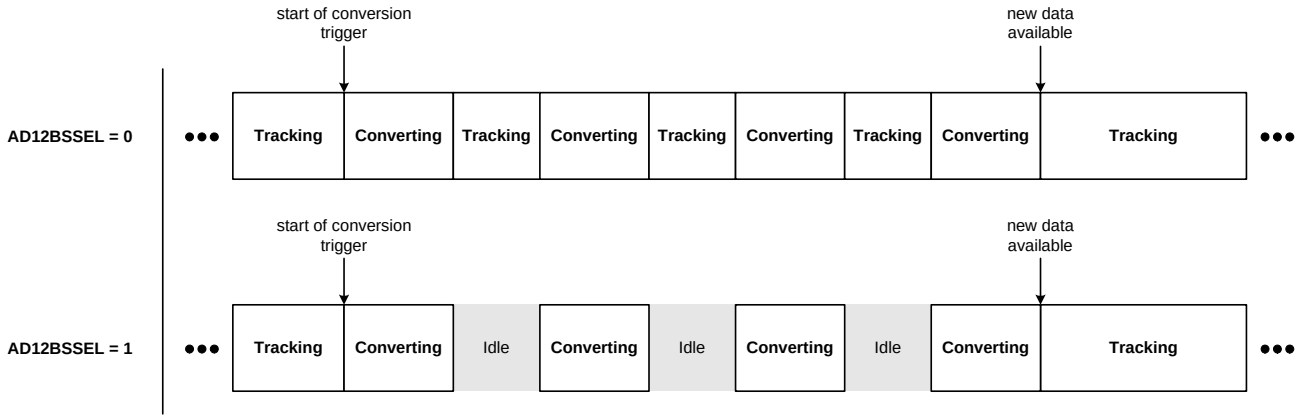


Figure 33.4. 12-bit Sampling Options

33.3. Burst Mode

The ADC implements a "burst mode" feature which enables lower power operation of the system. When a conversion trigger event occurs, the ADC will power on (if needed), track for a selected period of time, perform one or more conversions, and then return to the idle or powered-down state. The repeat counter (CHRxRPT) dictates the number of conversions performed for the channel being converted. Burst mode is enabled by setting the BURSTEN bit in the CONTROL register to 1.

In burst mode, the ADCEN bit controls the power consumption of the ADC between conversions. When ADCEN is cleared to 0, the ADC is powered down after each burst. If ADCEN is set to 1, the ADC will remain powered on between bursts.

In burst mode, the ADC can use a high-speed, low-power oscillator for conversion timing, enabling the system designer to run the core from a slow clock source or put the core in a low-power state to conserve power. Burst mode can also be configured to use the APB clock as a source. The clock used for burst mode is selected using the BCLKSEL bit in the CONFIG register.

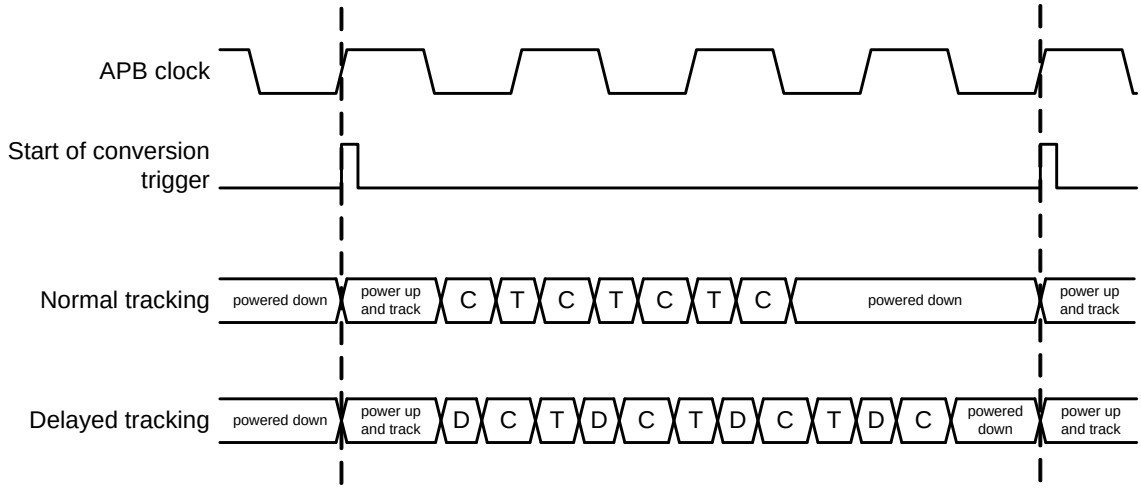
33.3.1. Data Accumulation

When burst mode is enabled, ADC samples can automatically be accumulated by the converter as they are taken. The accumulation mode is controlled by the ACCMD bit in the CONTROL register. This bit should be cleared to 0 to enable accumulation. Firmware must write the initial value (typically zero) to the ACC register prior to a conversion being taken or a scan sequence being initiated. The number of conversions accumulated is determined by the repeat counter field of the selected conversion characteristic register. For example, if CHxRPT is set to sample four times, four conversions will be accumulated.

When using the ADC's scan function, the accumulator will automatically be cleared to zero before the sequencer moves to the next selected channel. In other modes, this does not occur, and firmware must clear the ACC register, as necessary. Note that the ACC register should not be written while conversions are in progress, and its contents cannot be read.

33.3.2. Burst Mode Tracking

Tracking time in burst mode is different than non-burst mode operation of the ADC. For the first conversion, the tracking time is determined by the power-on time selected by the PWRTIME field in the CONTROL register. For all subsequent conversions during an ADC burst, the tracking time is dictated by the setting of the BMTK and the TRKMD bits. This timing is described in the BMTK bit description in the CONTROL register. Figure illustrates the burst mode tracking timing when ADCEN is cleared to 0 (power down between conversions), the burst clock is the low power oscillator, and the APB clock is operating at a low frequency.



C – converting
T – burst mode tracking time (BMTK)
D – delay time (TRKMD) (3 SAR clocks)

Burst Mode Tracking and Conversion Timing (Four Samples)

33.4. Channel Sequencer

The channel sequencer allows the user to define up to eight different combinations of ADC configurations and multiplexer channels, then scan through them with an ADC scan operation, relieving software of the overhead necessary to change multiplexer channels and other parameters.

33.4.1. Multiplexer Input Settings

The ADC module can select between multiple external and internal inputs. These inputs become the positive inputs to the single-ended SARADC module. The possible input selections for SARADC0 are shown in Table 33.2. Note that for some selections, other device circuitry must be enabled.

33.4.2. Timeslot Settings

The Channel Sequencer Time Slot Setup registers SQ7654 and SQ3210 configure the eight channel sequencer time slots each ADC. Each time slot defines the multiplexer channel to be converted, as well as which conversion characteristic to use for the conversion.

If the TSnMUX field in a time slot is set to the terminate scan value (0x1F), this indicates that no conversion is to be performed on this channel. When the sequencer encounters an 0x1F value in the time slot mux selection or converts the final time slot, it will either halt (single scan mode) or wrap back to time slot 0 (continuous scan mode).

The scan done interrupt flag (SDI in the STATUS register) will be set to 1 when a single scan operation is complete or when firmware clears SCANEN. Figure 33.5 shows a typical setup for a single scan using three sequencer time slots.

33.4.3. Conversion Characteristic Settings

The Conversion Characteristic Setup registers CHAR10 and CHAR32 define some of the characteristics of how the SARADC conversions will be performed. Each register defines two conversion characteristics (for example, CHAR10 defines conversion characteristic 1 and 0). The conversion characteristics select the gain, accumulation levels, post-conversion shifting, number of data bits, and whether to use the window comparator hardware. See the CHAR10 and CHAR32 register descriptions for more details on the selectable options.

33.4.4. Channel Scan Mode

The ADC implements channel sequencer logic which is capable of "scanning" through one or more ADC configurations automatically. When SCANEN is set to 1, the channel sequencer is active. The sequencer begins with time slot 0, and continues scanning through all eight time slots in sequence until it reaches the last channel or the final time slot or the terminate scan value (0x1F).

33.4.4.1. Single Scan Mode

If the SCANMD field in the CONFIG register is set to 0, the ADC performs a single scan through the channels. When using single-scan mode, each scan must be initiated by a 0-to-1 transition of the SCANEN bit in the CONFIG register. The scan will begin on the next conversion trigger event and end when the final channel in the sequencer has been converted. Note that a conversion trigger event is required for each channel in the scan, and the length of time between each conversion trigger event needs to be greater than or equal to the time required for the longest conversion. SCANEN will return to 0 upon completion of a scan in the single scan mode.

33.4.4.2. Continuous Scan Mode

If the SCANMD field in the CONFIG register is set to 1, the ADC will continue to wrap through all channels of the sequencer indefinitely. The continuous scan operation must be initiated by a 0-to-1 transition of the SCANEN bit in the CONFIG register. When the channel sequencer reaches the end of the sequence or the terminate scan value (0x1F), the ADC will begin again with the first channel. The scans will continue until firmware clears the SCANEN bit to 0.

33.4.4.3. Important notes on the use of Scan Mode

Note the following important restrictions on the use of Scan Mode:

1. Scan should not be used if interleaved or simultaneous modes are enabled.
2. Burst mode should be enabled (BURSTEN = 1) if using scan.
3. A start-of-conversion trigger is required for each time slot in the sequence. Each trigger needs to be spaced farther apart in time than the amount of time required for the longest conversion in the sequence.

For example, if one time slot uses a 64-sample accumulation, the length of time between each conversion needs to be longer than the time required for one 64-sample accumulation.

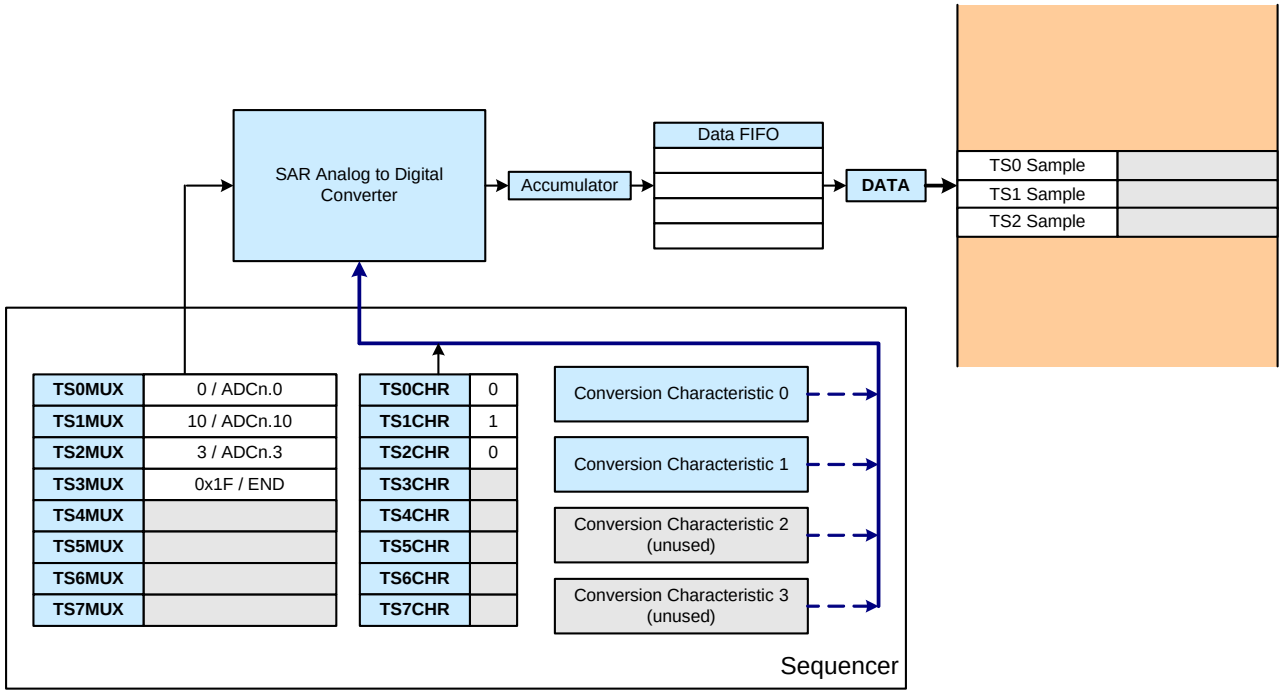


Figure 33.5. Channel Scan Setup and Timing

33.4.5. Single Channel Mode

When SCANEN is cleared to 0, the channel sequencer is not active. In this mode, time slot 0 is used for all conversions performed by the ADC. Any of the conversion characteristic setup registers can be used to define the conversion parameters in single configuration mode. The single conversion complete interrupt flag (SCCI in the STATUS register) will be set to 1 after each conversion is complete.

Setting the TSnMUX to 0x1F will terminate the scan at that time slot.

Table 33.2. SARADC0 Input Channels

SARADC0 Input	SARADC0 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
ADC0.0	Normal Input	PB0.4	PB0.3	PB0.3
ADC0.1	Normal Input	PB0.9	PB0.8	PB0.9
ADC0.2	Normal Input	PB0.10	PB0.9	PB2.0
ADC0.3	Normal Input	PB0.11	PB1.4	PB2.1
ADC0.4	Normal Input	PB1.4	PB1.5	PB2.2
ADC0.5	Normal Input	PB1.5	PB1.6	PB2.3
ADC0.6	Normal Input	PB1.6	PB2.0	PB2.4
ADC0.7	Normal Input	PB1.7	PB2.4	PB2.5
ADC0.8	Normal Input	PB2.0	PB2.5	PB2.6
ADC0.9	Normal Input	PB2.1	PB2.6	PB2.7
ADC0.10	Normal Input	PB2.4	PB2.7	PB3.3
ADC0.11	Normal Input	PB2.5	PB3.0	PB3.4
ADC0.12	Normal Input	PB2.6	PB3.1	PB3.5
ADC0.13	Normal Input	PB2.7	PB3.8	PB3.6
ADC0.14	Normal Input	PB3.0	PB3.9	PB3.7
ADC0.15	Normal Input	PB3.1	Reserved	PB3.8
ADC0.16	Normal Input	PB3.2	Reserved	PB3.9
ADC0.17	Normal Input	PB3.3	Reserved	Reserved
ADC0.18	Normal Input	PB3.12	Reserved	Reserved
ADC0.19	Normal Input	PB3.13	PB4.2	Reserved
ADC0.20	Normal Input	PB0.0	PB0.0	PB0.0
ADC0.21	Normal Input	PB0.1	Reserved	Reserved
ADC0.22	Normal Input	PB0.2	PB0.1	PB0.1
ADC0.23	Normal Input	PB0.3	PB0.2	PB0.2
ADC0.24	Internal Channel	VSS		
ADC0.25	Internal Channel	Digital LDO Output		
ADC0.26	Internal Channel	Memory LDO Output		
ADC0.27	Internal Channel	VDC		

Table 33.2. SARADC0 Input Channels (Continued)

SARADC0 Input	SARADC0 Input Description	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
ADC0.28	Internal Channel	VBAT		
ADC0.29	Internal Channel	1/2 Charge Pump Output		
ADC0.30	Internal Channel	Temperature Sensor Output		

Figure 33.6.

33.5. Voltage Reference Configuration

The ADC has the option to use several voltage reference sources: the on-chip VREF module, an external voltage reference, a dedicated internal 1.65 V (nominal) fast-settling reference for the SARADC block, the internal 1.8 V Analog LDO, or the VBAT voltage supply. The VREFSEL field in the CONTROL register selects the voltage reference for the ADC. The internal fast-settling reference will be automatically enabled if it is selected. Optionally, when using the external VREF pin as the reference source, the reference ground is selectable between an internal ground node tied to VSS and the external VREFGND pin. The REFGNDSEL bit in the CONTROL register determines which option is used. When REFGNDSEL is set to use VREFGND and an external input is being measured, the VREFGND pin signal is also used as the ADC's signal ground reference. Using the external VREFGND can provide for cleaner ADC conversions with an external reference source. The various VREF configuration options are shown in Figure 33.7.

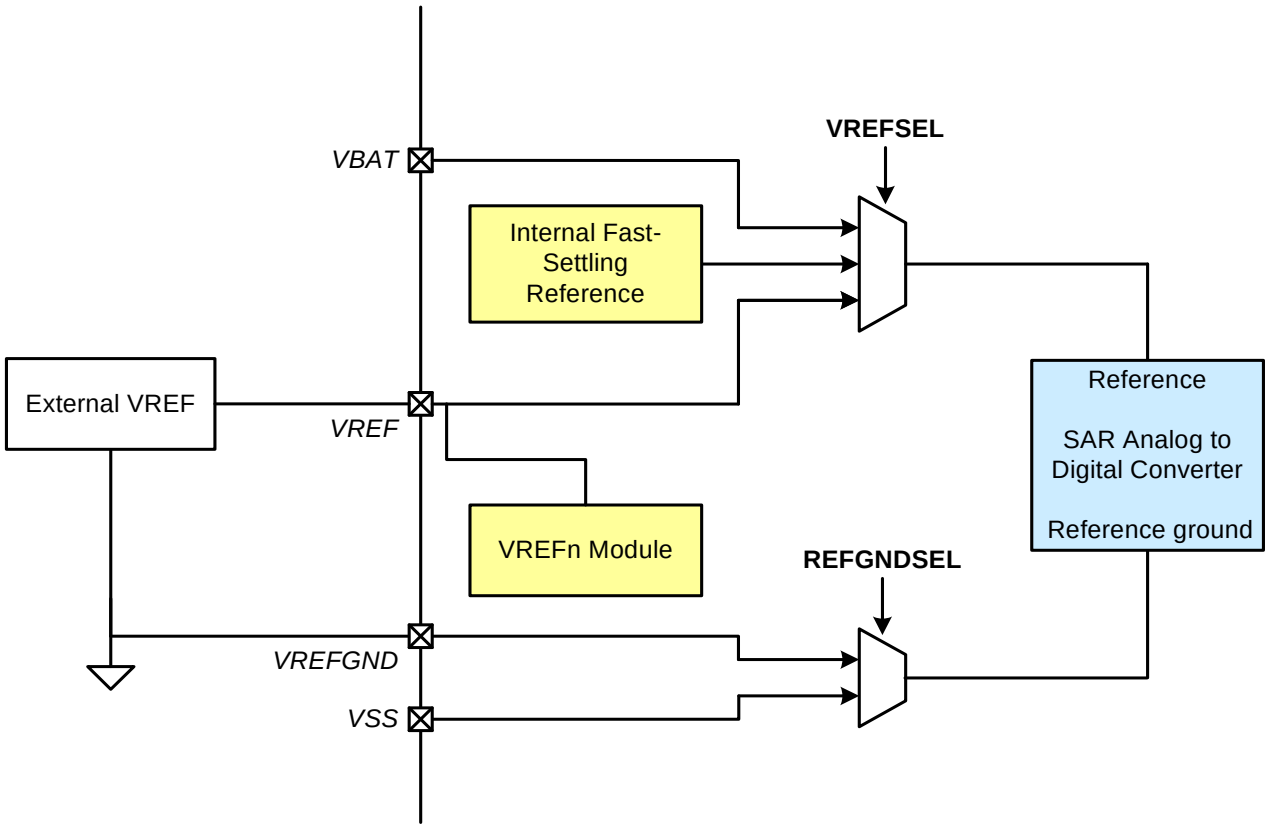


Figure 33.7. Voltage Reference Options

33.6. Power Configuration

When the ADC is disabled, it will remain in a powered-down, inactive state. The ADC is enabled by setting the ADCEN bit in the CONTROL register to 1, and there are several fields in the CONTROL register to reduce operational power when the ADC is enabled. The MREFLPEN bit can be used to reduce the power to the internal buffers when the SAR clock is operated at a slower speed. Additionally, the BIASSEL field has four selectable power levels that can scale power in regards to the SAR clock speed. The LPMDEN bit is used to reduce the current required during the tracking phase. If the application allows for longer tracking times, LPMDEN can be set to 1.

33.7. Data Output

The ADC allows for several different configurations and post-processing options on the output data. In the basic configuration, individual samples are written into a FIFO at the end of each conversion, and can be read through the DATA register or transferred to RAM using the DMA. The FIFOLVL field in the FIFOSTATUS register indicates the number of 32-bit words currently available in the FIFO. Each data word may contain one or two ADC samples. The order and number of the data words is selected by the PACKMD field in the CONFIG register. When the MSB of PACKMD is 0, only one sample is written per word, and the LSB of the PACKMD field determines whether the sample is written to the upper half or lower half of the 32-bit word. When the MSB of PACKMD is 1, two samples are packed into each output word. The order in which the samples are packed is determined by the LSB of PACKMD. The DPSTS bit in the FIFOSTATUS register indicates the target for the next sample within a data word (upper half or lower half). Figure 33.8 and Figure 33.9 show the data packing options.

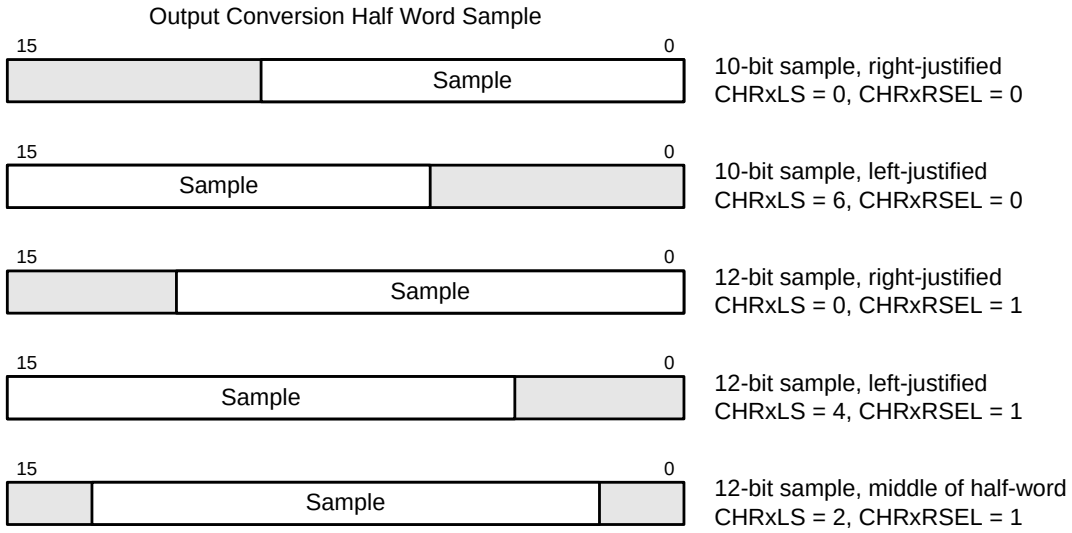


Figure 33.8. Sample Formatting

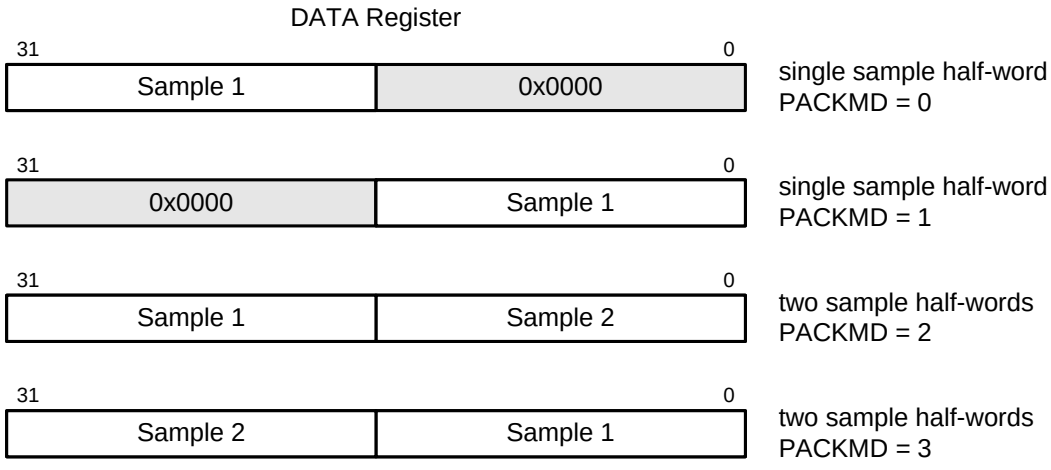


Figure 33.9. ADC Output Data Packing Options

33.7.1. Output Data Window Comparator

The ADC includes an output data window comparator. Using the window comparator, the ADC output data can be automatically compared against a specified upper and lower limit and trigger an interrupt, when desired. The window comparator limits are set by the WCLIMITS register. The WCGT field in WCLIMITS sets a "greater than" comparison limit, and the WCLT field sets a "less than" comparison limit. These two limit values are always compared against a right-justified output after any accumulation has been performed on the data. The window comparator is enabled for individual channel characteristics as detailed in “33.4. Channel Sequencer” .

The window comparator limits work together to determine when an interrupt will be triggered. Firmware can configure the ADC to generate an interrupt when the output is within the two limits or outside of the limits. To generate an interrupt within the two limits, WCLT should be programmed to a higher value than WCGT. To generate an interrupt outside of the two limits, WCGT should be programmed to a higher value than WCLT. Figure 33.10, Figure 33.11, Figure 33.12, and Figure 33.13 show examples of configuring the window comparator limits for different situations.

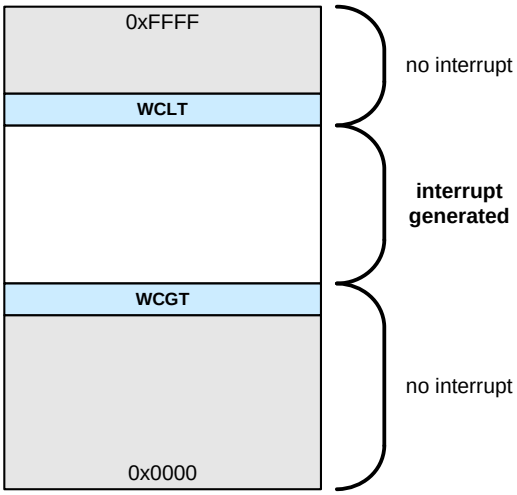


Figure 33.10. Example Window Comparator Limits for Inside Range (WCGT < WCLT)

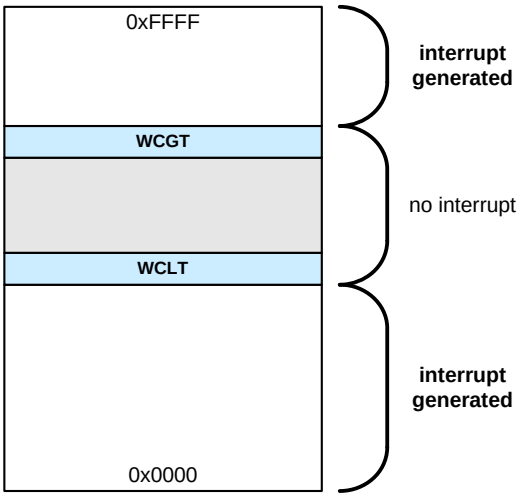


Figure 33.11. Example Window Comparator Limits for Outside Range (WCGT > WCLT)

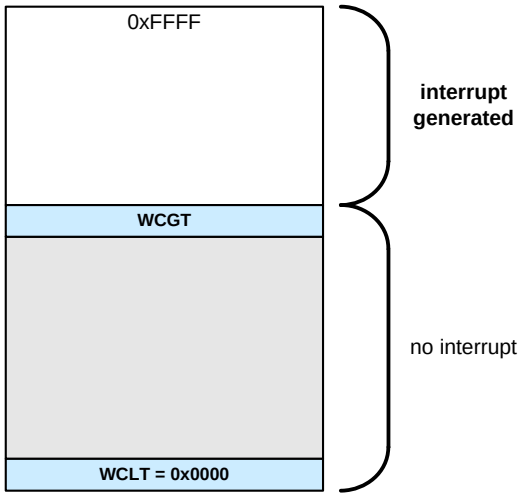


Figure 33.12. Example Window Comparator Limits for Above Value (WCGT = Value, WCLT = 0)

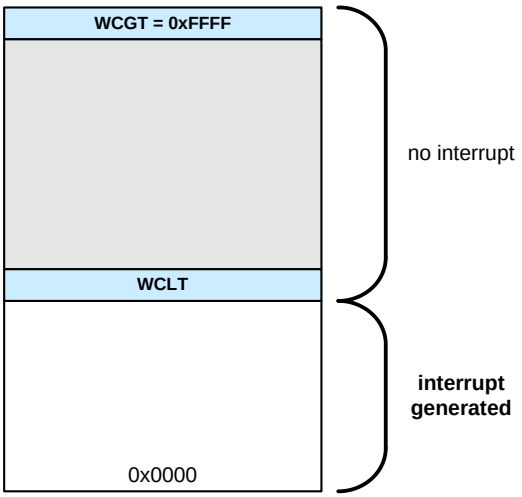


Figure 33.13. Example Window Comparator Limits for Below Value (WCLT = Value, WCGT = Full Scale)

33.8. Interrupts

The ADC interrupt can be triggered by five different, independently maskable interrupt sources. Two of these interrupts indicate error conditions, while the other three are primarily used for non-DMA operation. All interrupt status flags are located in the STATUS register and must be cleared by software. Descriptions of each interrupt condition are below:

- FURI: The FIFO underrun interrupt flag is set when a read from the DATA register is initiated while the FIFO is empty (FIFOLVL = 0). The read of the DATA register in this event will return the previous ADC result.
- FORI: The FIFO overrun interrupt flag is set when a new ADC data word (containing one or two samples) is to be written to the FIFO and the FIFO is full. If a FIFO overrun occurs, the data word that triggered the overrun will be lost.

- SDI: The scan done interrupt flag is set when scan mode is enabled and a channel scan sequence completes a single scan operation. In continuous scan mode, the SDI flag will be set when firmware exits scan.
- SCCI: The single conversion complete interrupt flag is set at the end of every conversion or accumulated conversion (when accumulation is enabled).
- WCI: The window comparator interrupt is set when a window comparison event happens and the current sequencer channel has enabled the interrupt.

33.9. DMA Configuration and Usage

DMA can be used to pipe the ADC data out of the FIFO into RAM, allowing more bandwidth for the core to perform other tasks. Note that the DMA will only start a transfer when there are 4 elements in the FIFO. For the SARADC module, the DMA should be configured as follows:

- Source size (SRCSIZE) and destination size (DSTSIZE) are 2 for a word transfer.
- The source address increment (SRCAIMD) is 3 for non-incrementing mode.
- The destination address increment (DSTAIMD) is 2 for word increments.
- NCOUNT (where NCOUNT+1 is the number of 4-byte words) and RPOWER (where 2^{RPOWER} is the number of data transfers) set as described below. Note that RPOWER > 2 is not valid setting.
  - RPOWER = 0 and NCOUNT = 0. As soon as the FIFO reaches 4 words, the first word will be transferred using the DMA. In this configuration, there will always be 3 lagging elements in the FIFO.
  - RPOWER = 1 and NCOUNT = 1. As soon as the FIFO reaches 4 words, the first two words will be transferred using the DMA. In this configuration, there will always be 2 lagging elements in the FIFO.
  - RPOWER = 2 and NCOUNT = 3. As soon as the FIFO reaches 4 words, all elements in the FIFO will be transferred using the DMA. In this configuration, there will be no lagging elements in the FIFO.

Once the DMA is configured, writing a 1 to DMAEN in the CONFIG register will enable the DMA transfers from the ADC. The FIFO will continue to be serviced by the DMA until the specified transfer operation is complete.

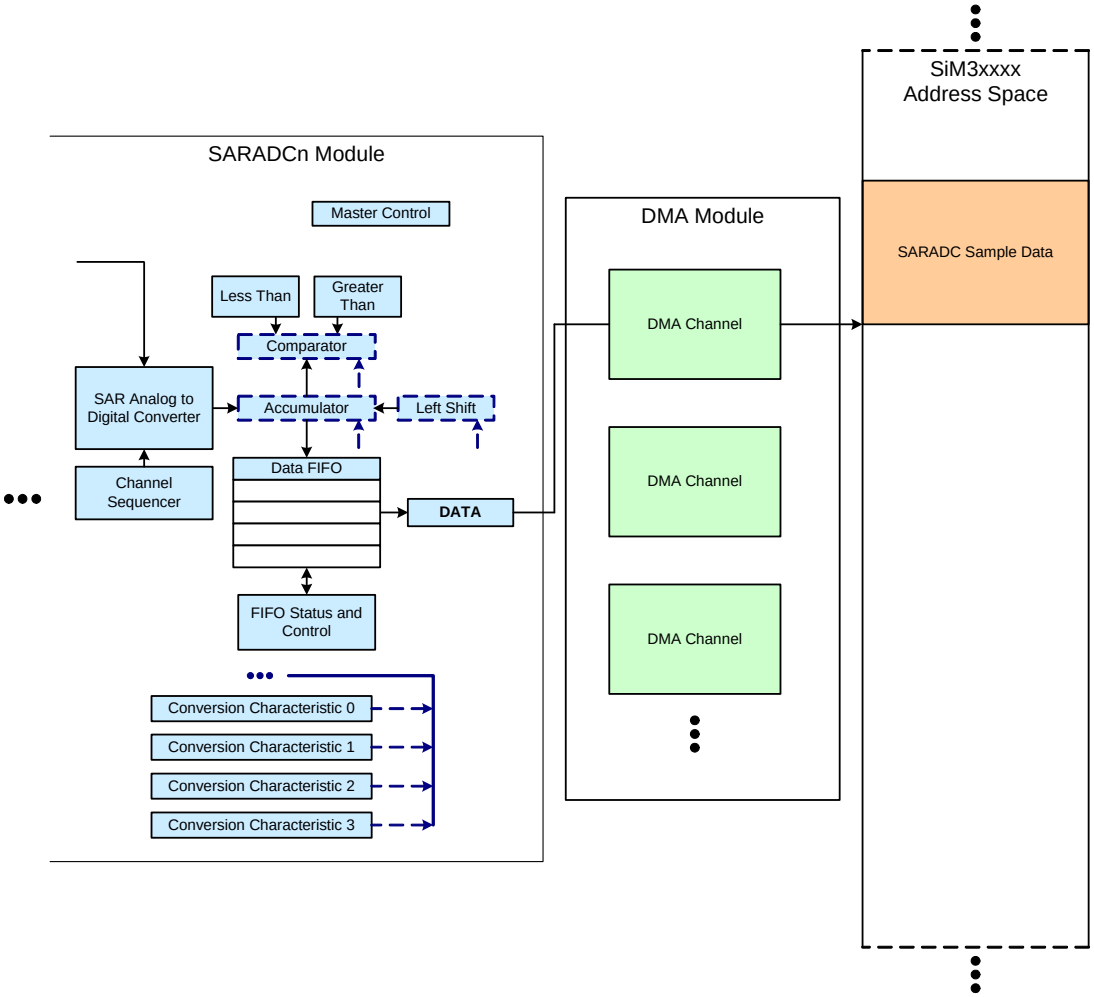


Figure 33.14. SAR ADC DMA Configuration

33.10. SARADC0 Registers

This section contains the detailed register descriptions for SARADC0 registers.

Register 33.1. SARADC0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	FURIEN	FORIEN	SDIEN	SCCIEN	CLKDIV										
Type	R	RW	RW	RW	RW	RW										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BCLKSEL	DMAEN	Reserved	SCANMD	Reserved	SCANEN	Reserved		PACKMD		Reserved					
Type	RW	RW	R	RW	R	RW	RW		RW		RW					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_CONFIG = 0x4001_A000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 33.3. SARADC0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	FURIEN	**FIFO Underrun Interrupt Enable.** 0: Disable the data FIFO underrun interrupt. 1: Enable the data FIFO underrun interrupt.
29	FORIEN	**FIFO Overrun Interrupt Enable.** 0: Disable the data FIFO overrun interrupt. 1: Enable the data FIFO overrun interrupt.
28	SDIEN	**Scan Done Interrupt Enable.** This bit enables the generation of an interrupt when the channel sequencer has cycled through all of the specified time slots. 0: Disable the ADC scan complete interrupt. 1: Enable the ADC scan complete interrupt.
27	SCCIEN	**Single Conversion Complete Interrupt Enable.** 0: Disable the ADC single data conversion complete interrupt. 1: Enable the ADC single data conversion complete interrupt.

Table 33.3. SARADC0_CONFIG Register Bit Descriptions

Bit	Name	Function
26:16	CLKDIV	**SAR Clock Divider.** This field sets the ADC clock divider value. It should be configured to be as close to the maximum SAR clock speed as the datasheet will allow. When CLKDIV < 3, the APB clock is used as the SAR clock. When CLKDIV ≥ 3, the SAR clock frequency is given by the following equation: $F_{CLKSAR} = \frac{2 \times F_{APB}}{CLKDIV + 1}$
15	BCLKSEL	**Burst Mode Clock Select.** 0: Burst mode uses the Low Power Oscillator. 1: Burst mode uses the APB clock.
14	DMAEN	**DMA Interface Enable .** 0: Disable the ADC module DMA interface. 1: Enable the ADC module DMA interface.
13	Reserved	Must write reset value.
12	SCANMD	**Scan Mode Select.** 0: The channel sequencer will cycle through all of the specified time slots once. 1: The channel sequencer will cycle through all of the specified time slots in a loop until SCANEN is cleared to 0.
11	Reserved	Must write reset value.
10	SCANEN	**Scan Mode Enable.** Setting this bit to 1 enables the ADC to scan through the specified time slots in the channel sequencer. The sequence begins on a 0-to-1 transition of this bit, so it must be 0 before a write to 1 to have any effect. 0: Disable ADC scan mode. 1: Enable ADC scan mode. The ADC will scan through the defined time slots in sequence on every start of conversion.
9:8	Reserved	Must write reset value.
7:6	PACKMD	**Output Packing Mode.** This field specifies how the ADC output data will be packed into the data registers. 00: Data is written to the upper half-word and the lower half-word is filled with 0's. An SCI interrupt is triggered when data is written, if enabled. 01: Data is written to the lower half-word, and the upper half-word is filled with 0's. An SCI interrupt is triggered when data is written, if enabled. 10: Two data words are packed into the register with the upper half-word representing the earlier data, and the lower half-word representing the later data. The ADC write to the lower half-word will trigger the SCI interrupt, if enabled. 11: Two data words are packed into the register with the lower half-word representing the earlier data, and the upper half-word representing the later data. The ADC write to the upper half-word will trigger the SCI interrupt, if enabled.
5:0	Reserved	Must write reset value.

Register 33.2. SARADC0_CONTROL: Measurement Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VREFSEL		Reserved		MREFLPEN	LPMDEN	BIASSEL		ADBUSY	TRKMD	ACCMD	Reserved	VCMEN	AD12BSSEL	ADCEN	BURSTEN
Type	RW		RW		RW	RW	RW		RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	PWRTIME				SCSEL				BMTK						CLKESEL	REFGNDSEL
Type	RW				RW				RW						RW	RW
Reset	1	1	1	1	0	0	0	0	0	1	1	1	1	0	0	0

Register ALL Access Address

SARADC0_CONTROL = 0x4001_A010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 33.4. SARADC0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:30	VREFSEL	**Voltage Reference Select.** 00: Select the internal, dedicated SARADC voltage reference as the ADC reference. 01: Select the VBAT pin as the ADC reference. 10: Select the output of the internal LDO regulator (~1.8 V) as the ADC reference. 11: Select the VREF pin as the ADC reference. This option is used for either an external VREF or the on-chip VREF driving out to the VREF pin.
29:28	Reserved	Must write reset value.
27	MREFLPEN	**MUX and VREF Low Power Enable.** This bit is used to limit the power of the internal buffers used on the reference and the mux. It can be set to 1 to reduce power consumptionwhen the SAR clock is slowed down. 0: Disable low power mode. 1: Enable low power mode (SAR clock $\leq$ 4 MHz).

Table 33.4. SARADC0_CONTROL Register Bit Descriptions

Bit	Name	Function
26	LPMDEN	**Low Power Mode Enable.** This bit can be used to reduce power to one of the ADC's internal nodes. It can be set to 1 to reduce power when tracking times in the application are longer (slower sample rates). 0: Disable low power mode. 1: Enable low power mode (requires extended tracking time).
25:24	BIASSEL	**Bias Power Select.** This field can be used to adjust the ADC's power consumption based on the conversion speed. Higher bias currents allow for faster conversion times. 00: Select bias current mode 0. Recommended to use modes 1, 2, or 3. 01: Select bias current mode 1 (SARCLK = 16 MHz). 10: Select bias current mode 2. 11: Select bias current mode 3 (SARCLK = 4 MHz).
23	ADBUSY	**ADC Busy.** This bit indicates that the ADC is currently converting (not tracking). Writing 1 to this bit in "On Demand" trigger mode initiates a conversion.
22	TRKMD	**ADC Tracking Mode.** 0: Normal Tracking Mode: When the ADC is enabled, a conversion begins immediately following the start-of-conversion signal. 1: Delayed Tracking Mode: When the ADC is enabled, a conversion begins 3 SAR clock cycles following the start-of-conversion signal. The ADC is allowed to track during this time.
21	ACCMD	**Accumulation Mode.** This bit is used to enable or disable accumulation when burst mode is enabled (BURSTEN = 1). 0: Conversions will be accumulated for the specified number of cycles in burst mode according to the channel configuration. 1: Conversions will not be accumulated in burst mode.
20	Reserved	Must write reset value.
19	VCMEN	**Common Mode Buffer Enable.** 0: Disable the common mode buffer. 1: Enable the common mode buffer.
18	AD12BSSEL	**12-Bit Mode Sample Select.** This bit defines how the ADC samples the analog input in 12-bit mode. 0: The ADC re-samples the input before each of the four conversions. 1: The ADC samples once before the first conversion and converts four times.
17	ADCEN	**ADC Enable.** 0: Disable the ADC (low-power shutdown). 1: Enable the ADC (active and ready for data conversions).
16	BURSTEN	**Burst Mode Enable.** This bit should be set to enable burst mode or any other ADC mode which relies on burst mode (such as 12-bit operation).

Table 33.4. SARADC0_CONTROL Register Bit Descriptions

Bit	Name	Function
15:12	PWRTIME	**Burst Mode Power Up Time.** This field sets the time delay allowed for the ADC to power up from a low power state. $T_{PWRTIME} = \frac{8 \times PWRTIME}{F_{APB}}$
11:8	SCSEL	**Start-Of-Conversion Source Select.** This field specifies the event used to initiate conversions. 0000: An ADC conversion triggers from the ADCnT0 trigger source. 0001: An ADC conversion triggers from the ADCnT1 trigger source. 0010: An ADC conversion triggers from the ADCnT2 trigger source. 0011: An ADC conversion triggers from the ADCnT3 trigger source. 0100: An ADC conversion triggers from the ADCnT4 trigger source. 0101: An ADC conversion triggers from the ADCnT5 trigger source. 0110: An ADC conversion triggers from the ADCnT6 trigger source. 0111: An ADC conversion triggers from the ADCnT7 trigger source. 1000: An ADC conversion triggers from the ADCnT8 trigger source. 1001: An ADC conversion triggers from the ADCnT9 trigger source. 1010: An ADC conversion triggers from the ADCnT10 trigger source. 1011: An ADC conversion triggers from the ADCnT11 trigger source. 1100: An ADC conversion triggers from the ADCnT12 trigger source. 1101: An ADC conversion triggers from the ADCnT13 trigger source. 1110: An ADC conversion triggers from the ADCnT14 trigger source. 1111: An ADC conversion triggers from the ADCnT15 trigger source.
7:2	BMTK	**Burst Mode Tracking Time.** This field sets the time delay between consecutive conversions performed in Burst Mode. $T_{BMTK} = \frac{64 - BMTK + (3 \times TRKMD)}{F_{APB}}$ Note: The Burst Mode track delay is not inserted prior to the first conversion. The required tracking time for the first conversion should be defined with the ADPWM field.
1	CLKESEL	**Sampling Clock Edge Select.** This bit selects which edge of the APB clock is used during sampling. Note that if the core is halted and a conversion completes while this bit is set to 1, the SCCI bit will not set. It is recommended to leave this bit at 0 when debugging SAR-related firmware. 0: Select the rising edge of the APB clock. 1: Select the falling edge of the APB clock.

Table 33.4. SARADC0_CONTROL Register Bit Descriptions

Bit	Name	Function
0	REFGNDSEL	**Reference Ground Select.** This bit selects the reference ground for ADC conversions. The internal ground is always used for temperature sensor measurements. 0: The internal device ground is used as the ground reference for ADC conversions. 1: The VREFGND pin is used as the ground reference for ADC conversions.

Register 33.3. SARADC0_SQ7654: Channel Sequencer Time Slots 4-7 Setup

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	TS7MUX					TS7CHR		Reserved	TS6MUX					TS6CHR	
Type	R	RW					RW		R	RW					RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	TS5MUX					TS5CHR		Reserved	TS4MUX					TS4CHR	
Type	R	RW					RW		R	RW					RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_SQ7654 = 0x4001_A020

Table 33.5. SARADC0_SQ7654 Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30:26	TS7MUX	**Time Slot 7 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
25:24	TS7CHR	**Time Slot 7 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 7. 01: Select conversion characteristic 1 for time slot 7. 10: Select conversion characteristic 2 for time slot 7. 11: Select conversion characteristic 3 for time slot 7.
23	Reserved	Must write reset value.
22:18	TS6MUX	**Time Slot 6 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
17:16	TS6CHR	**Time Slot 6 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 6. 01: Select conversion characteristic 1 for time slot 6. 10: Select conversion characteristic 2 for time slot 6. 11: Select conversion characteristic 3 for time slot 6.

Table 33.5. SARADC0_SQ7654 Register Bit Descriptions

Bit	Name	Function
15	Reserved	Must write reset value.
14:10	TS5MUX	**Time Slot 5 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
9:8	TS5CHR	**Time Slot 5 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 5. 01: Select conversion characteristic 1 for time slot 5. 10: Select conversion characteristic 2 for time slot 5. 11: Select conversion characteristic 3 for time slot 5.
7	Reserved	Must write reset value.
6:2	TS4MUX	**Time Slot 4 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
1:0	TS4CHR	**Time Slot 4 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 4. 01: Select conversion characteristic 1 for time slot 4. 10: Select conversion characteristic 2 for time slot 4. 11: Select conversion characteristic 3 for time slot 4.

Register 33.4. SARADC0_SQ3210: Channel Sequencer Time Slots 0-3 Setup

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	TS3MUX					TS3CHR		Reserved	TS2MUX					TS2CHR	
Type	R	RW					RW		R	RW					RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	TS1MUX					TS1CHR		Reserved	TS0MUX					TS0CHR	
Type	R	RW					RW		R	RW					RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_SQ3210 = 0x4001_A030

Table 33.6. SARADC0_SQ3210 Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30:26	TS3MUX	**Time Slot 3 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
25:24	TS3CHR	**Time Slot 3 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 3. 01: Select conversion characteristic 1 for time slot 3. 10: Select conversion characteristic 2 for time slot 3. 11: Select conversion characteristic 3 for time slot 3.
23	Reserved	Must write reset value.
22:18	TS2MUX	**Time Slot 2 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
17:16	TS2CHR	**Time Slot 2 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 2. 01: Select conversion characteristic 1 for time slot 2. 10: Select conversion characteristic 2 for time slot 2. 11: Select conversion characteristic 3 for time slot 2.

Table 33.6. SARADC0_SQ3210 Register Bit Descriptions

Bit	Name	Function
15	Reserved	Must write reset value.
14:10	TS1MUX	**Time Slot 1 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot.
9:8	TS1CHR	**Time Slot 1 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 1. 01: Select conversion characteristic 1 for time slot 1. 10: Select conversion characteristic 2 for time slot 1. 11: Select conversion characteristic 3 for time slot 1.
7	Reserved	Must write reset value.
6:2	TS0MUX	**Time Slot 0 Input Channel.** A value of x in this field selects the ADCn.x channel as the ADCn input during this time slot. Set this field to 0x1F to terminate the scan at this slot. Time Slot 0 is also used to specify the parameters for single (non-scan mode) conversions.
1:0	TS0CHR	**Time Slot 0 Conversion Characteristic.** Selects which of the conversion characteristic settings is used for this time slot. 00: Select conversion characteristic 0 for time slot 0. 01: Select conversion characteristic 1 for time slot 0. 10: Select conversion characteristic 2 for time slot 0. 11: Select conversion characteristic 3 for time slot 0.

Register 33.5. SARADC0_CHAR32: Conversion Characteristic 2 and 3 Setup

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved							CHR3WCIEN	CHR3RSEL	CHR3LS			CHR3RPT			CHR3GN
Type	R							RW	RW	RW			RW			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved							CHR2WCIEN	CHR2RSEL	CHR2LS			CHR2RPT			CHR2GN
Type	R							RW	RW	RW			RW			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_CHAR32 = 0x4001_A040

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 33.7. SARADC0_CHAR32 Register Bit Descriptions

Bit	Name	Function
31:25	Reserved	Must write reset value.
24	CHR3WCIEN	**Conversion Characteristic 3 Window Comparator Interrupt Enable.** Enable window comparison interrupts for this channel. 0: Disable window comparison interrupts. 1: Enabled window comparison interrupts. The window comparator will be used to check the ADC result on channels that use this characteristic.
23	CHR3RSEL	**Conversion Characteristic 3 Resolution Selection.** Select between 10- and 12-bit mode. 0: Select 10-bit Mode. 1: Select 12-bit Mode (burst mode must be enabled).
22:20	CHR3LS	**Conversion Characteristic 3 Left-Shift Bits.** This field specifies the number of bits to shift the result left at conversion completion. A zero value produces a fully right-justified result.

Table 33.7. SARADC0_CHAR32 Register Bit Descriptions

Bit	Name	Function
19:17	CHR3RPT	**Conversion Characteristic 3 Repeat Counter.** This field determines the number of samples the converter will accumulate in burst mode when the accumulation option is enabled. 000: Accumulate one sample. 001: Accumulate four samples. 010: Accumulate eight samples. 011: Accumulate sixteen samples. 100: Accumulate thirty-two samples (10-bit mode only). 101: Accumulate sixty-four samples (10-bit mode only). 110-111: Reserved.
16	CHR3GN	**Conversion Characteristic 3 Gain.** 0: The on-chip PGA gain is 1. 1: The on-chip PGA gain is 0.5.
15:9	Reserved	Must write reset value.
8	CHR2WCIEN	**Conversion Characteristic 2 Window Comparator Interrupt Enable.** Enable window comparison interrupts for this channel. 0: Disable window comparison interrupts. 1: Enabled window comparison interrupts. The window comparator will be used to check the ADC result on channels that use this characteristic.
7	CHR2RSEL	**Conversion Characteristic 2 Resolution Selection.** Select between 10- and 12-bit mode. 0: Select 10-bit Mode. 1: Select 12-bit Mode (burst mode must be enabled).
6:4	CHR2LS	**Conversion Characteristic 2 Left-Shift Bits.** This field specifies the number of bits to shift the result left at conversion completion. A zero value produces a fully right-justified result.
3:1	CHR2RPT	**Conversion Characteristic 2 Repeat Counter.** This field determines the number of samples the converter will accumulate in burst mode when the accumulation option is enabled. 000: Accumulate one sample. 001: Accumulate four samples. 010: Accumulate eight samples. 011: Accumulate sixteen samples. 100: Accumulate thirty-two samples (10-bit mode only). 101: Accumulate sixty-four samples (10-bit mode only). 110-111: Reserved.
0	CHR2GN	**Conversion Characteristic 2 Gain.** 0: The on-chip PGA gain is 1. 1: The on-chip PGA gain is 0.5.

Register 33.6. SARADC0_CHAR10: Conversion Characteristic 0 and 1 Setup

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved							CHR1WCIEN	CHR1RSEL	CHR1LS			CHR1RPT			CHR1GN
Type	R							RW	RW	RW			RW			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved							CHR0WCIEN	CHR0RSEL	CHR0LS			CHR0RPT			CHR0GN
Type	R							RW	RW	RW			RW			RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_CHAR10 = 0x4001_A050

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 33.8. SARADC0_CHAR10 Register Bit Descriptions

Bit	Name	Function
31:25	Reserved	Must write reset value.
24	CHR1WCIEN	**Conversion Characteristic 1 Window Comparator Interrupt Enable.** Enable window comparison interrupts for this channel. 0: Disable window comparison interrupts. 1: Enabled window comparison interrupts. The window comparator will be used to check the ADC result on channels that use this characteristic.
23	CHR1RSEL	**Conversion Characteristic 1 Resolution Selection.** Select between 10- and 12-bit mode. 0: Select 10-bit Mode. 1: Select 12-bit Mode (burst mode must be enabled).
22:20	CHR1LS	**Conversion Characteristic 1 Left-Shift Bits.** This field specifies the number of bits to shift the result left at conversion completion. A zero value produces a fully right-justified result.

Table 33.8. SARADC0_CHAR10 Register Bit Descriptions

Bit	Name	Function
19:17	CHR1RPT	**Conversion Characteristic 1 Repeat Counter.** This field determines the number of samples the converter will accumulate in burst mode when the accumulation option is enabled. 000: Accumulate one sample. 001: Accumulate four samples. 010: Accumulate eight samples. 011: Accumulate sixteen samples. 100: Accumulate thirty-two samples (10-bit mode only). 101: Accumulate sixty-four samples (10-bit mode only). 110-111: Reserved.
16	CHR1GN	**Conversion Characteristic 1 Gain.** 0: The on-chip PGA gain is 1. 1: The on-chip PGA gain is 0.5.
15:9	Reserved	Must write reset value.
8	CHR0WCIEN	**Conversion Characteristic 0 Window Comparator Interrupt Enable.** Enable window comparison interrupts for this channel. 0: Disable window comparison interrupts. 1: Enabled window comparison interrupts. The window comparator will be used to check the ADC result on channels that use this characteristic.
7	CHR0RSEL	**Conversion Characteristic 0 Resolution Selection.** Select between 10- and 12-bit mode. 0: Select 10-bit Mode. 1: Select 12-bit Mode (burst mode must be enabled).
6:4	CHR0LS	**Conversion Characteristic 0 Left-Shift Bits.** This field specifies the number of bits to shift the result left at conversion completion. A zero value produces a fully right-justified result.
3:1	CHR0RPT	**Conversion Characteristic 0 Repeat Counter.** This field determines the number of samples the converter will accumulate in burst mode when the accumulation option is enabled. 000: Accumulate one sample. 001: Accumulate four samples. 010: Accumulate eight samples. 011: Accumulate sixteen samples. 100: Accumulate thirty-two samples (10-bit mode only). 101: Accumulate sixty-four samples (10-bit mode only). 110-111: Reserved.
0	CHR0GN	**Conversion Characteristic 0 Gain.** 0: The on-chip PGA gain is 1. 1: The on-chip PGA gain is 0.5.

Register 33.7. SARADC0_DATA: Output Data Word

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	R															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

SARADC0_DATA = 0x4001_A060

Table 33.9. SARADC0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**Output Data Word.** The DATA register represents the oldest information available in the FIFO. When DATA is read, FIFOLVL decrements by one, and the FIFO pointer will point to the next value in the FIFO. Data is packed according to the PACKMD field.

Register 33.8. SARADC0_WCLIMITS: Window Comparator Limits

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WCGT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	WCLT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_WCLIMITS = 0x4001_A070

Table 33.10. SARADC0_WCLIMITS Register Bit Descriptions

Bit	Name	Function
31:16	WCGT	**Greater-Than Window Comparator Limit.** This field is the right-justified "greater than" parameter for the window comparator. ADC output data will be compared against this value when it is available. When WCLT is greater than WCGT, an ADC result between the two limits will cause a window compare interrupt, if enabled. When WCLT is less than WCGT, an ADC result above or below the two limits (but not in between) will cause a window compare interrupt, if enabled.
15:0	WCLT	**Less-Than Window Comparator Limit.** This field is the right-justified "less than" parameter for the window comparator. ADC output data will be compared against this value when it is available. When WCLT is greater than WCGT, an ADC result between the two limits will cause a window compare interrupt, if enabled. When WCLT is less than WCGT, an ADC result above or below the two limits (but not in between) will cause a window compare interrupt, if enabled.

Register 33.9. SARADC0_ACC: Accumulator Initial Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	ACC															
Type	W															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_ACC = 0x4001_A080

Table 33.11. SARADC0_ACC Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	ACC	**Accumulator Initial Value.** This write-only field is used to set the accumulator to an initial value. In most cases, this field should be written to zero before beginning a conversion or a scan sequence when accumulation is enabled (ACCMD = 0).

Register 33.10. SARADC0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											FURI	FORI	SDI	SCCI	WCI
Type	R											RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

SARADC0_STATUS = 0x4001_A090

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 33.12. SARADC0_STATUS Register Bit Descriptions

Bit	Name	Function
31:5	Reserved	Must write reset value.
4	FURI	**FIFO Underrun Interrupt.** This bit is set to 1 by hardware when a FIFO underrun event has occured, and can be used to trigger an interrupt if enabled. This bit must be cleared by software.
3	FORI	**FIFO Overrun Interrupt.** This bit is set to 1 by hardware when a FIFO overrun event has occured, and can be used to trigger an interrupt if enabled. This bit must be cleared by software.
2	SDI	**Scan Done Interrupt.** This bit is set to 1 by hardware when a scan operation is complete, and can be used to trigger an interrupt if enabled. This bit must be cleared by software.
1	SCCI	**Single Conversion Complete Interrupt.** This bit is set to 1 by hardware at the end of each conversion, and can be used to trigger an interrupt if enabled. This bit must be cleared by software.
0	WCI	**Window Compare Interrupt.** This bit is set to 1 by hardware when a window comparator event has occurred, and can be used to trigger an interupt if enabled. This bit must be cleared by software.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 33.11. SARADC0_FIFOSTATUS: FIFO Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved										DRDYF	DPSTS	FIFOLVL			
Type	R										R	R	R			
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Register ALL Access Address

SARADC0_FIFOSTATUS = 0x4001_A0A0

Table 33.13. SARADC0_FIFOSTATUS Register Bit Descriptions

Bit	Name	Function
31:6	Reserved	Must write reset value.
5	DRDYF	**Data Ready Flag.** This bit indicates that new data is ready to be written into the FIFO. It is set after the data conversion is complete and cleared by any new conversion start trigger. 0: New data is not produced yet. 1: New data is ready.
4	DPSTS	**Data Packing Status.** This is a read only status bit indicating to which half-word the hardware will write the next ADC output data. 0: The next ADC conversion will be written to the lower half-word. 1: The next ADC conversion will be written to the upper half-word.
3:0	FIFOLVL	**FIFO Level.** This is the number of ADC words in the FIFO. Each word may contain one or two samples depending on the packing mode (PACKMD).

33.11. SARADC0 Register Memory Map

Table 33.14. SARADC0 Memory Map

SARADC0_SQ3210	SARADC0_SQ7654	SARADC0_CONTROL	SARADC0_CONFIG	Register Name
0x4001_A030	0x4001_A020	0x4001_A010	0x4001_A000	ALL Address
ALL	ALL	ALL | SET | CLR	ALL | SET | CLR	Access Methods
Reserved	Reserved	VREFSEL	Reserved	Bit 31
TS3MUX	TS7MUX		FURIEN	Bit 30
		Reserved	FORIEN	Bit 29
			SDIEN	Bit 28
		MREFLPEN	SCCIEN	Bit 27
		LPMDEN	CLKDIV	Bit 26
TS3CHR	TS7CHR	BIASSEL		Bit 25
				Bit 24
Reserved	Reserved	ADBUSY		Bit 23
TS2MUX	TS6MUX	TRKMD		Bit 22
		ACCMD		Bit 21
		Reserved		Bit 20
		VCMEN		Bit 19
		AD12BSSEL		Bit 18
TS2CHR	TS6CHR	ADCEN		Bit 17
		BURSTEN		Bit 16
Reserved	Reserved	PWRTIME	BCLKSEL	Bit 15
TS1MUX	TS5MUX		DMAEN	Bit 14
			Reserved	Bit 13
TS1CHR	TS5CHR	SCSEL	SCANMD	Bit 12
			Reserved	Bit 11
			SCANEN	Bit 10
			Reserved	Bit 9
				Bit 8
Reserved	Reserved	BMTK	PACKMD	Bit 7
TS0MUX	TS4MUX			Bit 6
			Reserved	Bit 5
				Bit 4
				Bit 3
				Bit 2
TS0CHR	TS4CHR	CLKESEL		Bit 1
		REFGNDSEL		Bit 0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 33.14. SARADC0 Memory Map

Register Name	SARADC0_WCLIMITS	SARADC0_DATA	SARADC0_CHAR10	SARADC0_CHAR32	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ALL Address	0x4001_A070	0x4001_A060	0x4001_A050	0x4001_A040																																
Access Methods	ALL	ALL	ALL | SET | CLR	ALL | SET | CLR																																
	WCGT	DATA	Reserved	Reserved								CHR3WCIEN	CHR3RSEL	CHR3LS		CHR3RPT		CHR3GN									CHR2WCIEN	CHR2RSEL	CHR2LS		CHR2RPT			CHR2GN	
	WCLT											CHR1WCIEN	CHR1RSEL	CHR1LS		CHR1RPT		CHR1GN									CHR0WCIEN	CHR0RSEL	CHR0LS		CHR0RPT			CHR0GN	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 33.14. SARADC0 Memory Map

Register Name	SARADC0_ACC	SARADC0_STATUS	SARADC0_FIFOSTATUS
ALL Address	0x4001_A080	0x4001_A090	0x4001_A0A0
Access Methods	ALL	ALL | SET | CLR	ALL
Bit 31	Reserved	Reserved	Reserved
Bit 30			
Bit 29			
Bit 28			
Bit 27			
Bit 26			
Bit 25			
Bit 24			
Bit 23			
Bit 22			
Bit 21			
Bit 20			
Bit 19			
Bit 18			
Bit 17			
Bit 16			
Bit 15	ACC		
Bit 14			
Bit 13			
Bit 12			
Bit 11			
Bit 10			
Bit 9			
Bit 8			
Bit 7			
Bit 6			
Bit 5			DRDYF
Bit 4		FURI	DPSTS
Bit 3		FORI	FIFOLVL
Bit 2		SDI	
Bit 1		SCCI	
Bit 0		WCI	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

34. Serial Peripheral Interface (SPI0 and SPI1)

This section describes the Serial Peripheral Interface (SPI) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the SPI block, which is used by SPI0 and SPI1 on all device families covered in this document.

34.1. SPI Features

The SPI module includes the following features:

- Supports 3- or 4-wire master or slave modes.
- Supports up to 10 MHz clock in master mode and one-tenth of the APB clock in slave mode.
- Support for all clock phase polarity and slave select (NSS) polarity modes.
- Hardware control of NSS.
- 16-bit programmable clock rate.
- Programmable MSB-first or LSB-first shifting.
- 8-byte FIFO buffers for both transmit and receive data paths to support high speed transfers.
- Programmable FIFO threshold level to request data service for DMA transfers.
- Support for multiple masters on the same data lines.
- Hardware flow control options (SPI1).

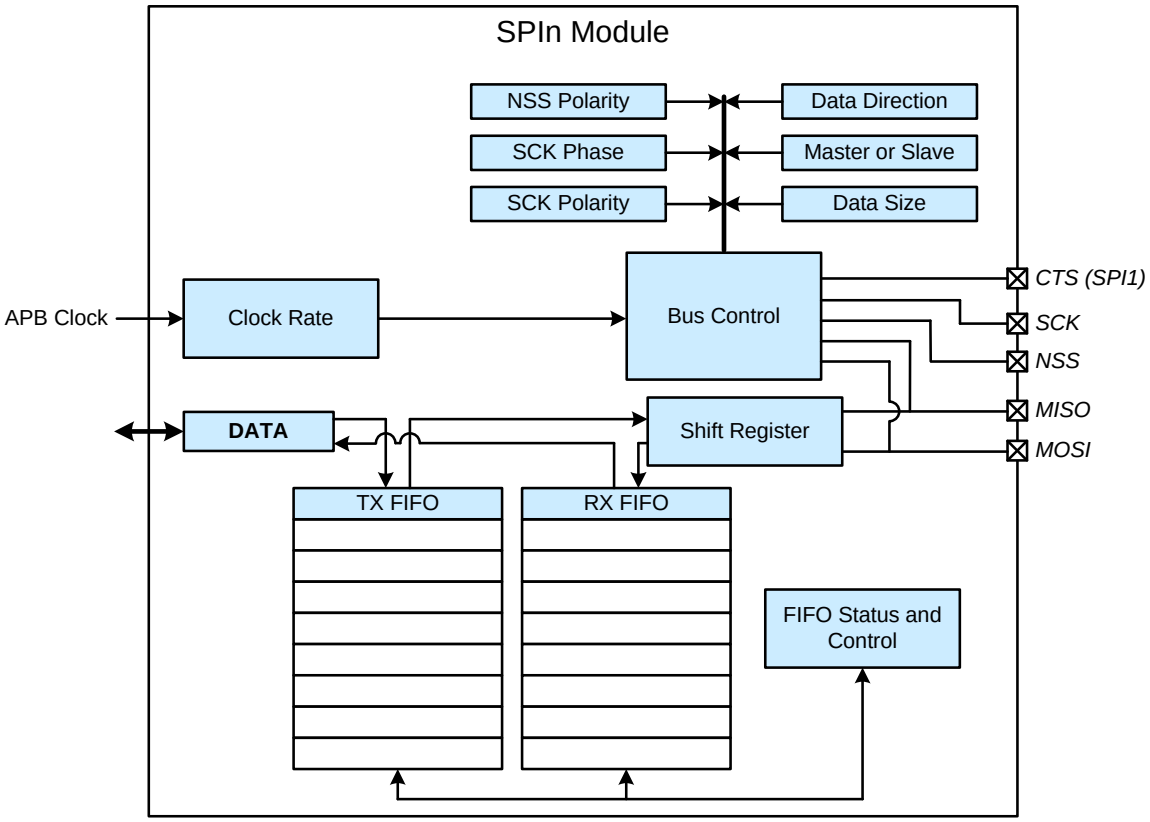


Figure 34.1. SPI Block Diagram

34.2. Signal Descriptions

The four signals used by the SPI module are MOSI, MISO, SCK, NSS. Figure 34.2 shows a typical SPI transfer. SPI1 also includes a handshaking signal (CTS). SPI0 is assigned to pins using the crossbar, while the SPI1 peripheral is available only on dedicated PB2 port I/O pins. To enable the SPI on the port, the SPI1SEL bit in register PBCFG0_CONTROL1 should be set to 1. SPI1 signals are mapped as shown in Table 34.1.

Table 34.1. SPI1 Pin Mapping

SPI1 Signal	Pin Name (All Packages)
SPI1_CTS	PB2.0
SPI1_SCLK	PB2.4
SPI1_MISO	PB2.5
SPI1_MOSI	PB2.6
SPI1_NSS	PB2.7

34.2.1. Master Out, Slave In (MOSI)

The master-out, slave-in (MOSI) signal is an output from a master device and an input to the slave devices on the bus. It is used to serially transfer data from the master to the slaves. This MOSI pin is an output when the module operates as a master and an input when operating as a slave.

34.2.2. Master In, Slave Out (MISO)

The master-in, slave-out (MISO) signal is an output from a slave device and an input to the master device. It is used to serially transfer data from the slave to the master. The MISO pin is an input when the SPI module operates as a master and an output when operating as a slave. The hardware places the MISO pin in a high-impedance state when the module is disabled or when the module operates in 4-wire mode as a slave that is not selected.

34.2.3. Serial Clock (SCK)

The serial clock (SCK) signal is an output from the master device and an input to the slave devices. It is used to synchronize the transfer of data between the master and slave on the MOSI and MISO pins. The SCK pin is an output driving the clock when operating as a master and an input receiving the clock when operating as a slave.

34.2.4. Slave Select (NSS)

The slave select (NSS) signal can be an output from a master device (in 4-wire single master mode), an input to a master device (in 4-wire multiple master), an input to a slave device (in 4-wire slave mode), or unused/ unconnected (in 3-wire master or 3-wire slave mode). The slave select mode (NSSMD) field in the CONFIG register configures the NSS pin for the desired mode.

The NSS signal may be optionally connected to a physical pin. The device port configuration module has more information.

34.2.5. Clear to Send (CTS–SPI1 Only)

The clear to send (CTS) signal is an input to the SPI1 module. When used in flow control mode, or transmit on request mode, the CTS signal may be used by a slave device to hold off or request transfers from the master.

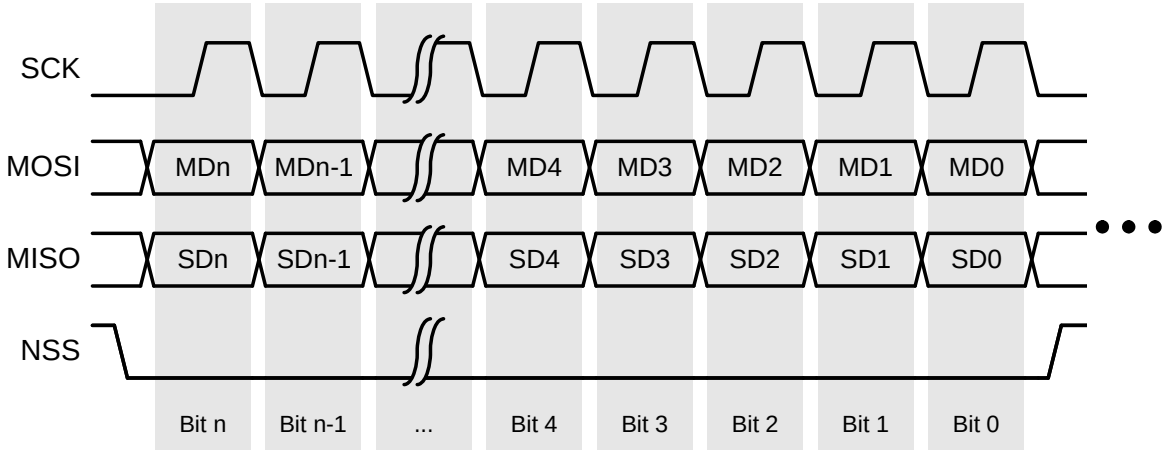


Figure 34.2. 4-Wire SPI Transfer

34.3. Clocking

The APB clock is the clock source for the SPI module. The SCK output clock in master mode is a divided version of this clock. In both master and slave modes, the clock signal present on the SCK pin determines when data is shifted out of or into the data shift register.

34.3.1. Master Mode Clocking

In master mode, an internal clock rate generator is used to divide down the APB clock to produce the desired SCK rate, and the hardware drives the SCK pin as an output from the device. The 16-bit clock divider (CLKDIV) field in the CLKRATE register sets the SCK frequency as a fraction of the APB clock. The CLKDIV field description describes the equation for the SCK frequency as a function of the APB clock.

34.3.2. Slave Mode Clocking

The CLKDIV field is not used in slave mode, and the SCK pin becomes an input to the device. A different device should be configured as the SPI bus master and is expected to drive the SCK input at the desired frequency. The maximum input SCK rate in slave mode is equal to the APB clock frequency divided by 10.

34.4. Signal Format

The SPI module has flexible data formatting options. The data length, clock phase, clock polarity, shift direction, and NSS polarity are all selectable via fields in the CONFIG register.

34.4.1. Data Size and Shift Direction

The data size (DSIZE) field configures the transfer data length to be between 1 and 16 bits. DSIZE should be set to one less than the desired data length; for an 8-bit data length, firmware should set DSIZE to 7.

The data direction select (DDIRSEL) field configures the data shift direction for both transmitted and received data. The hardware can shift data MSB first (DDIRSEL = 0) or LSB first (DDIRSEL = 1).

34.4.2. Slave Select Polarity

The slave select polarity (NSSPOL) bit determines the polarity of the NSS pin for both master mode where NSS is an output and for slave mode where NSS is an input. The pin can be active low (NSSPOL = 0) or active high (NSSPOL = 1).

34.4.3. Clock Phase and Polarity Configuration

The CLKPHA and CLKPOL bits configure the SCK pin phase and polarity, respectively. Clearing CLKPHA to 0 places the SCK rising or falling edge at the center of the data bit, and setting CLKPHA to 1 places the SCK rising or falling edge at the data bit transition edge. The clock can also be idle low (CLKPOL = 0) or idle high (CLKPOL = 1).

The CLKPHA and CLKPOL bits must be set in both master and slave modes. In master mode, these bits determine the characteristics of the clock driven on of the SCK output. In slave mode, these bits set the characteristics of the clock that the module expects to receive on the SCK input. The clock phase and polarity settings determine when the data transitions take place with respect to the SCK phase.

Figure 34.3 illustrates all clock polarity and phase combinations when DDIRSEL is cleared to 0. Figure 34.4 illustrates all clock polarity and phase combinations when DDIRSEL is set to 1.

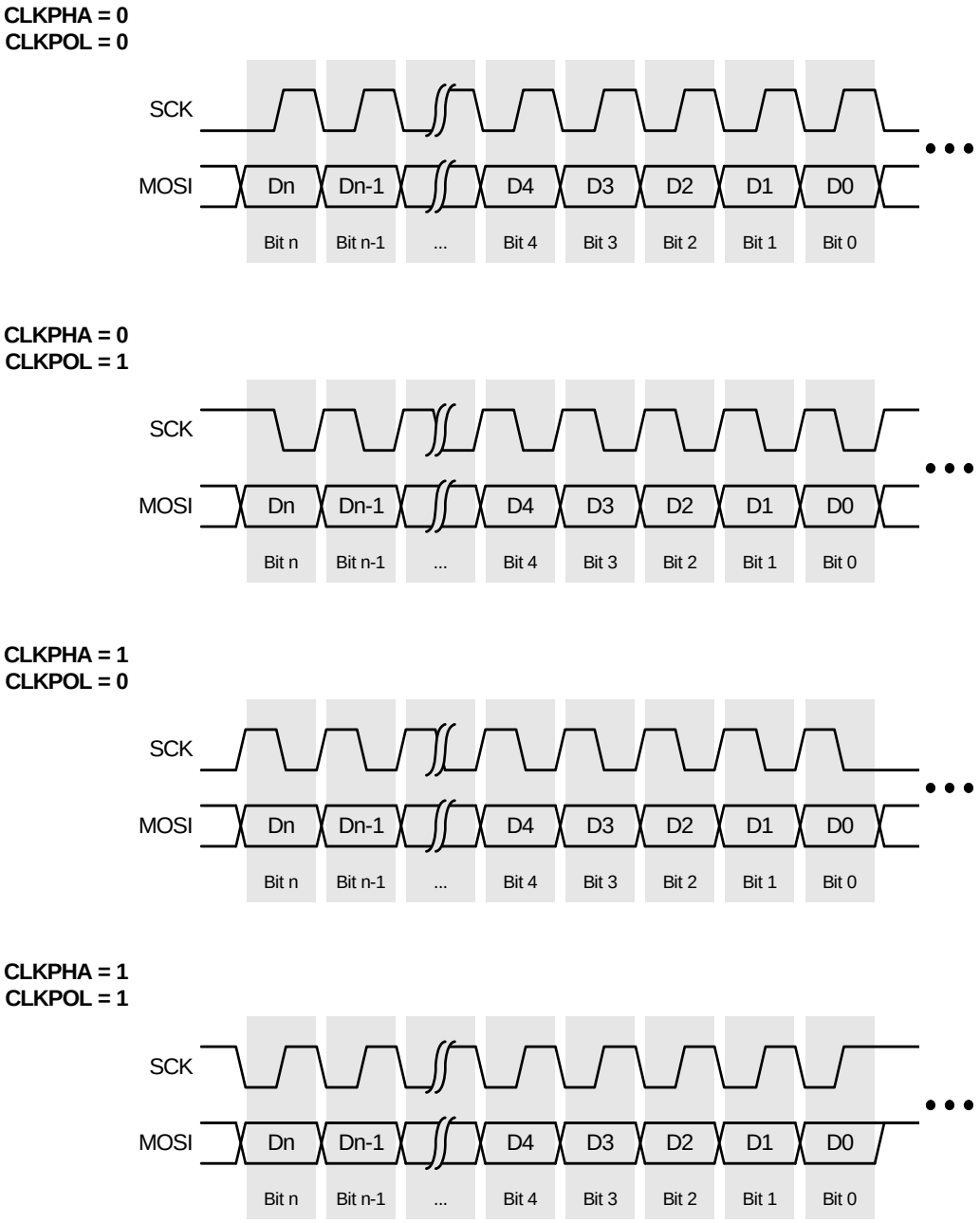


Figure 34.3. SPI Clock Polarity and Phase Combinations (DDIRSEL = 0, Master Mode)

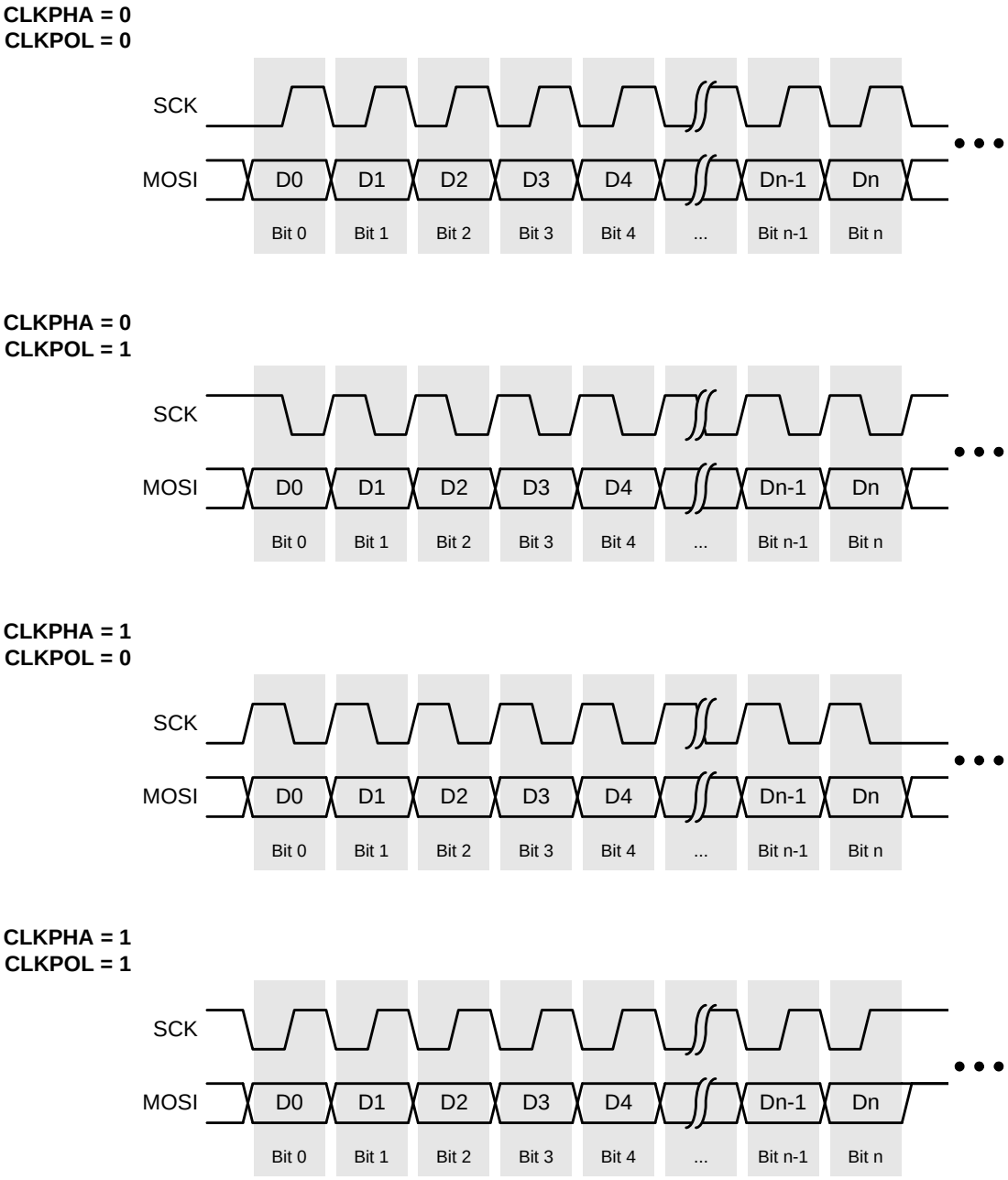


Figure 34.4. SPI Clock Polarity and Phase Combinations (DDIRSEL = 1, Master Mode)

34.5. Master Mode Configurations and Data Transfer

Firmware can set the MSTEN bit to 1 to configure the module as a master. The module supports three different options of master mode operation.

34.5.1. 3-Wire Single Master Mode

In 3-wire single master mode (NSSMD = 0), the slave select (NSS) pin is not required and need not be connected to port pins by the device port configuration module. In this mode, the device should be connected to a single slave device that either doesn't support an NSS input. To support multiple slaves in 3-wire single master mode, each slave device must have an NSS input, the NSS inputs must be connected to a unique port pin on the master device, and each pin must be controlled from firmware.

Figure 34.5 illustrates the master-slave connection diagram for 3-wire single master mode.

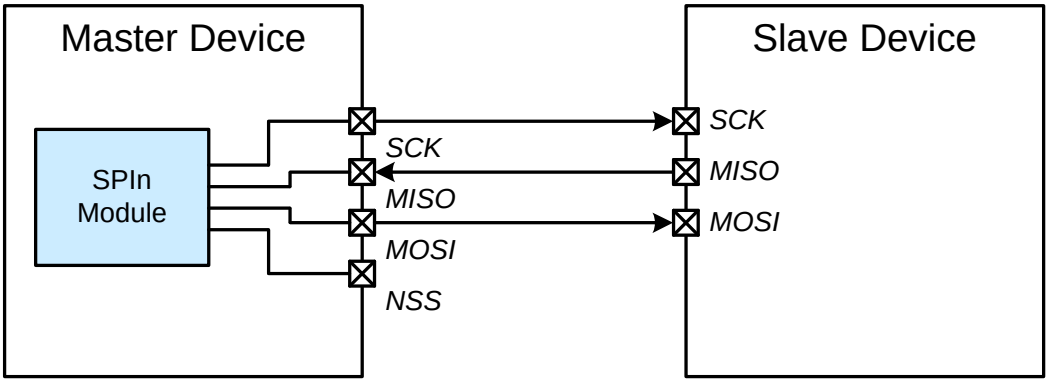


Figure 34.5. 3-Wire Single Master Mode Connection Diagram

34.5.2. 4-Wire Single Master Mode

In 4-wire single master mode (NSSMD = 2 or 3), the slave select (NSS) pin is configured as an output and should be connected to the NSS input of the first slave device. Any additional slave devices added on the SPI bus should have their NSS input driven by a firmware-controlled port pin output from the master device.

Figure 34.6 shows the master-slave connection diagram for 4-wire single master mode.

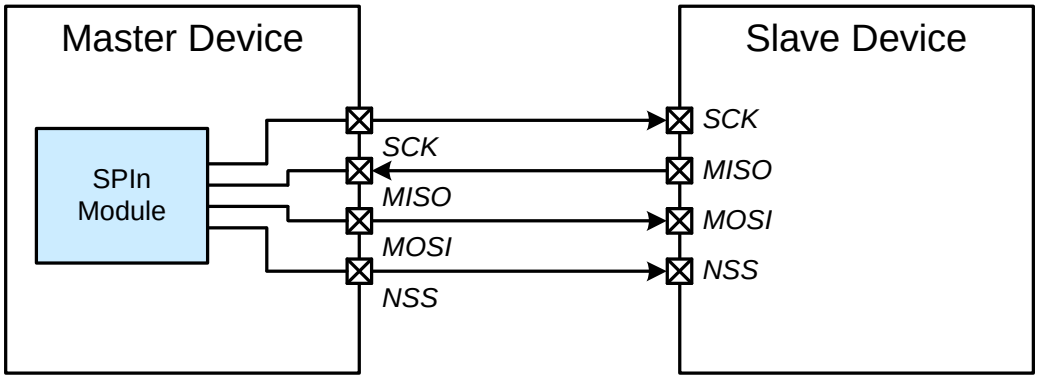


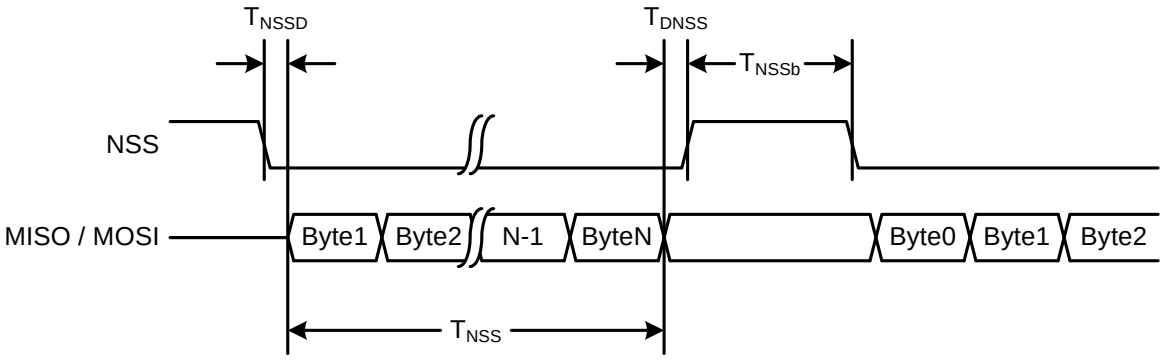
Figure 34.6. 4-Wire Single Master Mode Connection Diagram

34.5.2.1. NSS Control

By default, the least-significant bit of NSSMD controls the state of the master's NSS pin in 4-wire single-master mode. When the AUTONSS bit in the CONFIGMD register is set to 1, the master will automatically control the NSS pin. Timing of the NSS pin is controlled by the NSSCNT and NSSDELAY fields in the CONFIGMD register. Figure 34.7 shows the NSS timing relationship.

The NSSCNT field defines the number of SPI transfers to assert NSS. NSS will assert 1/2 SPI clock before beginning a data transfer, then proceed to perform NSSCNT+1 byte transfers. When the defined number of bytes have been transferred, NSS will remain asserted for an additional 1/2 SPI clock before de-asserting. If the SPI stalls before the defined number of bytes have been transferred (for example during a transmit FIFO underrun event), NSS will remain asserted, and the SPI will continue the transfer when the stall condition is lifted.

For back-to-back transfers, NSS is left de-asserted for a minimum of 1/2 SPI clock. The NSSDELAY field determines the amount of additional time provided between NSS assertions (in increments of 1/2 SPI clock). Thus the total amount of time which NSS will be de-asserted between blocks is equal to (NSSDELAY+1)/2 SPI clocks.



$T_{NSSD} = T_{DNSS} = ½$ SPI clock

T_{NSS} = NSSCNT+1 Bytes

T_{NSSb} = (NSSDELAY+1) x ½ SPI clocks

Figure 34.7. NSS Timing with AUTONSS = 1

34.5.3. 4-Wire Multiple Master Mode

Set the NSSMD (Slave Select Mode) field to 0x01 to configure the NSS pin for 4-wire slave / multi master mode.

In 4-wire multiple master mode (NSSMD = 1), the slave select (NSS) pin is configured as an input and is used to disable the SPI module while another SPI master accesses the bus. When the NSS input is driven low by the bus master, two events occur:

1. Hardware clears the master mode enable (MSTEN) and SPI enable (SPIEN) bits to disable the SPI module. The module must be manually re-enabled by firmware.
2. Hardware sets the mode fault interrupt (MDFI) flag. This will generate an interrupt if the mode fault interrupt is enabled (MDFIEN = 1).

Slave devices on the SPI bus should have their NSS inputs driven by a firmware-controlled port pin output from the master devices.

Figure 34.8 illustrates the connection diagram for 4-wire multiple master mode.

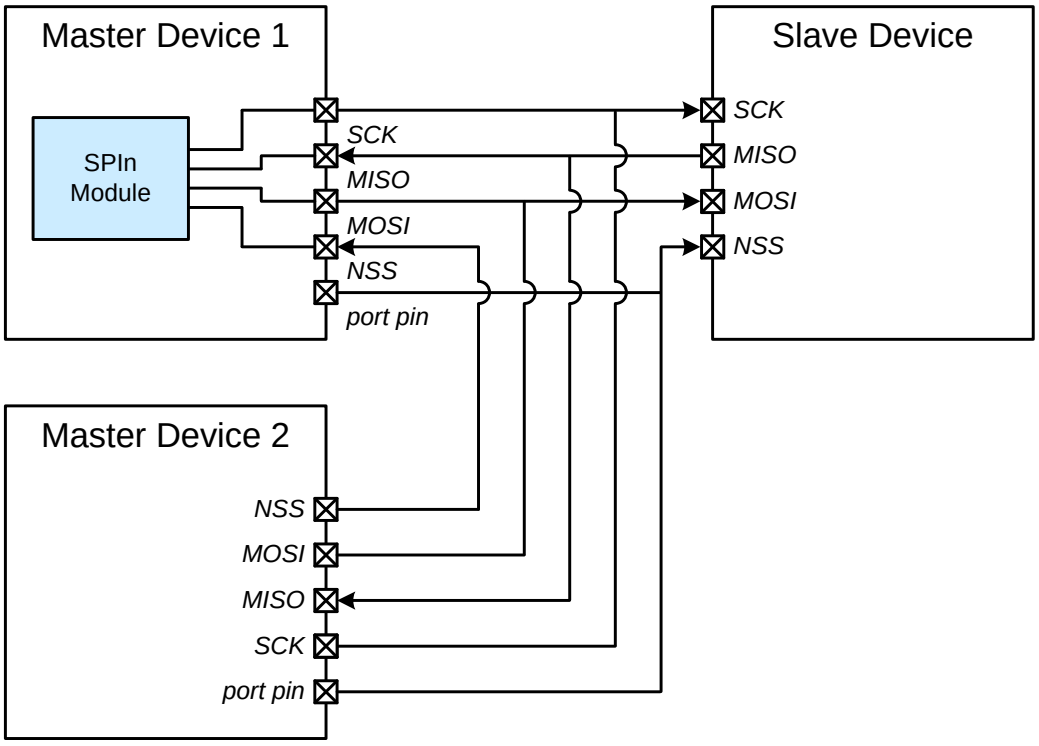


Figure 34.8. 4-Wire Multiple Master Mode Connection Diagram

34.5.4. Master Mode Data Transfer

A master device initiates all data transfers on a SPI bus. When the SPI module is first enabled by setting the SPIEN bit to 1, the transmit and receive FIFOs are empty. The hardware sets the transmit FIFO write request interrupt (TFRQI) flag when the number of empty slots in the transmit FIFO is at or above the transmit FIFO threshold (TFTH), which causes an interrupt, if enabled (TFRQIEN = 1). Firmware can then write to the DATA register in right-justified bytes, half-words, or full words to transfer data to the transmit FIFO.

When the shift register is empty, the hardware retrieves data from the transmit FIFO and begins a transmission. The module immediately shifts the data out serially on the MOSI line while driving the serial clock on SCK. When both the transmit FIFO and the shift register are empty, the hardware sets the underrun interrupt (URI) flag, resulting in an interrupt if the underrun interrupt is enabled (URIEN = 1).

The SPI bus is full-duplex, so the addressed SPI slave device may also be simultaneously transferring the contents of its shift register back to the SPI master using the MISO line as the master transfers data to a slave using MOSI. The shift register empty interrupt (SREI) flag serves as both a transmit-complete flag and receive-data-ready flag. When a received byte is fully transferred into the shift register, the hardware automatically moves it into the receive FIFO where it may be accessed by reading the DATA register.

34.6. Slave Mode Configurations and Data Transfer

Clearing the master mode enable (MSTEN) bit configures the SPI module for slave mode. Firmware should first fully configure the module (data length, clock polarity and phase, and slave mode) before setting the SPIEN bit to initiate SPI operations.

34.6.1. 3-Wire Slave Mode

In 3-wire slave mode (NSSMD = 0), the slave select (NSS) pin is not required and may not be connected to physical pins by the device's port configuration module. Because there is no means to uniquely address multiple slave devices in this mode, the device's SPI module should be the only slave device present on the bus in 3-wire slave mode. In addition, the bus signals should be in an idle state before enabling the module in 3-wire slave mode, since any unexpected transitions on SCK can cause erroneous bits to be shifted into the shift register without the NSS signal to gate the clock on SCK.

Figure 34.9 illustrates the connection diagram for 3-wire slave mode.

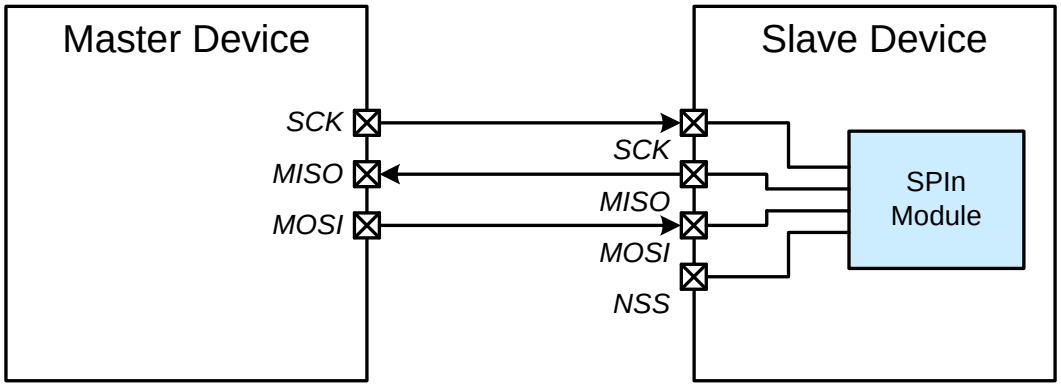


Figure 34.9. 3-Wire Single Slave Mode Connection Diagram

34.6.1.1. 4-Wire Slave Mode

In 4-wire slave mode (NSSMD = 2), the slave select (NSS) pin is configured as an input and should be connected to an NSS control signal from the master device.

Asserting the NSS signal enables the SPI module, and deasserting NSS disables the module. The polarity of the NSS input can be set using the NSSPOL bit. The NSS signal must be asserted for at least two APB clocks before the first active edge of SCK for each byte transfer.

Figure 34.10 shows the connection diagram for 4-wire slave mode.

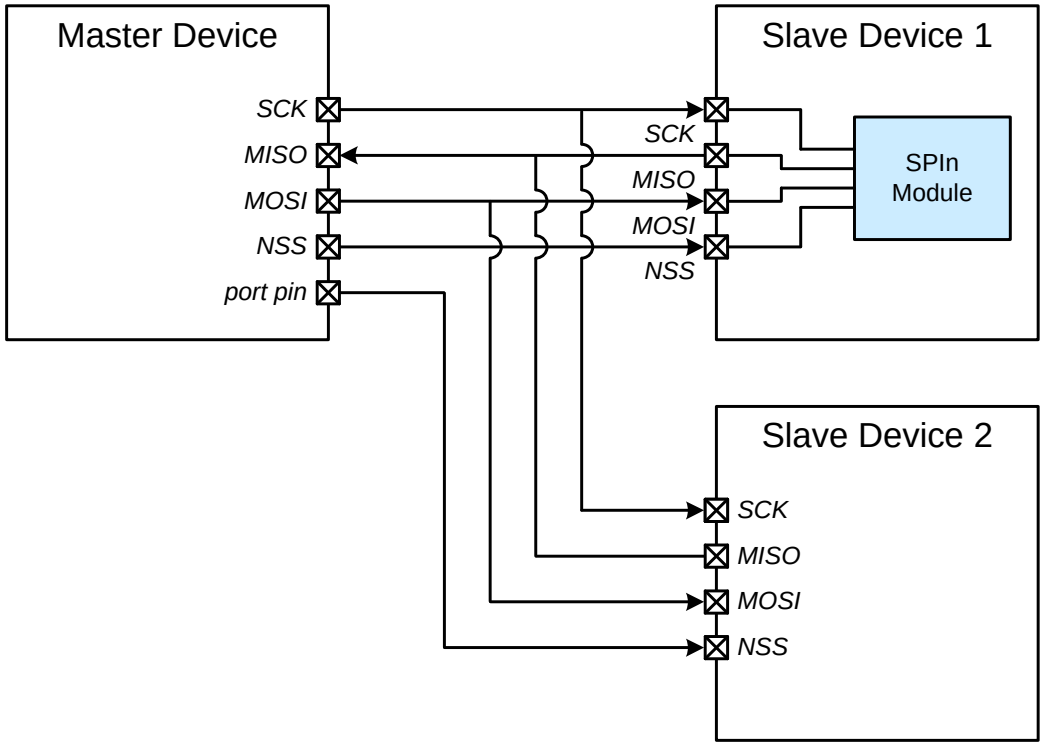


Figure 34.10. 4-Wire Slave Mode Connection Diagram

34.6.2. Slave Mode Data Transfer

In a SPI slave device, the master-generated clock input on the slave's SCK signal drives the data bytes received on MOSI and transmitted out through MISO. The hardware copies the data into the receive FIFO after receiving the number of bits specified by the DSIZE field. When the number of filled slots in the receive FIFO reaches or exceeds the programmed receive FIFO threshold (RFTH), hardware sets the receive FIFO read request interrupt (RFRQI) flag and generates an interrupt if RFRQIEN is set to 1. Firmware can read from the DATA register in right-justified bytes, half-words, or full words to pop data from the receive FIFO.

A slave device cannot initiate transfers and should pre-load the data into the transmit FIFO by writing to the DATA register before the master begins a transfer. If the shift register is empty, the hardware moves the first data in the transmit FIFO to the shift register in preparation for the data transfer.

34.7. Special Operation Modes and Functions

By default, the SPI operates as specified in sections 34.5 and 34.6. The CONFIGMD register adds additional functionality to the SPI module which allows for more autonomous operation of the peripheral.

34.7.1. Receive-Only Mode

When the OPMD field is set to 01b, the SPI operates in receive-only mode. Any data present in the FIFO will be transmitted. Once the FIFO is empty, the SPI will continue to receive data (in master mode, this means it will also continue to generate clocks). The SPI will transmit 0's if there is no data in the FIFO to transmit, but an underrun (URI flag) will not be generated when the transmit FIFO goes empty. If the receive FIFO becomes full, the SPI will stall, and stop generating clocks until the FIFO has been read and there is room for more data. Receive-only mode is terminated by disabling the SPI or by setting the ABORT bit. If ABORT is used, the SPI will complete the current data transfer and then disable itself.

34.7.2. Transmit-Only Mode

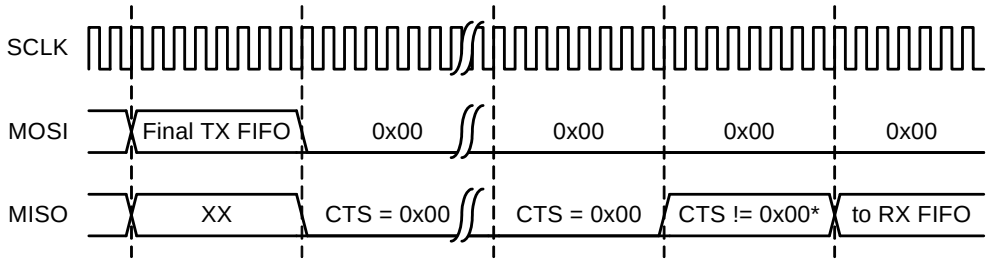
When the OPMD field is set to 10b, the SPI operates in transmit-only mode. In this mode, data will be transmitted, but any received data will be ignored and not stored in the FIFO. The SPI will stall and set the URI flag if the transmit FIFO goes empty. Transfers will continue when additional data becomes available. Transmit-only mode is terminated by disabling the SPI or by setting the ABORT bit. If ABORT is used, the SPI will complete the current data transfer and then disable itself.

34.7.3. Master Flow Control Mode

When OPMD is set to 11b and the MSTEN bit is set to 1, the SPI operates in flow control mode. As a master, the SPI will send any data in the transmit FIFO, until the FIFO goes empty. Depending on the state of the CTSEN bit, the SPI will either use data polling or pin polling to determine when additional data is available (see "34.7.3.1. CTS Polling" for details on CTS polling). When data becomes available, the SPI will transmit 0x00 and store the received data until firmware disables the SPI or uses the ABORT bit to end the transfer. If ABORT is used, the SPI will complete the current data transfer and then disable itself.

34.7.3.1. CTS Polling

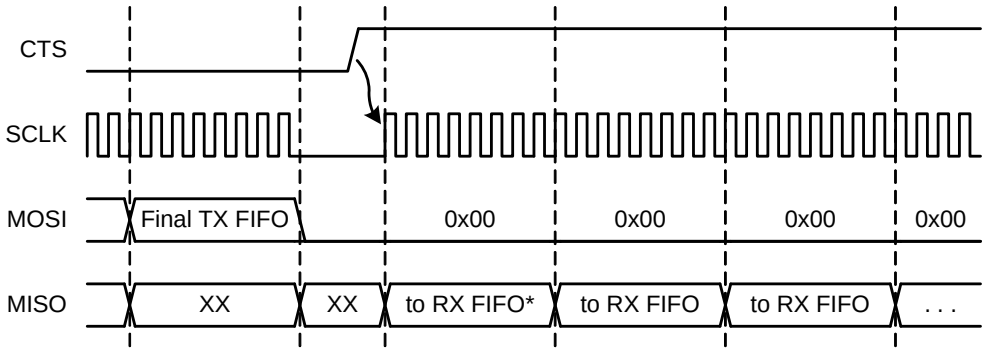
There are two CTS polling options available. If CTSEN is cleared to 0, data polling is used, as shown in Figure 34.11. During the polling phase, the SPI will send a single byte of 0x00 to the slave and then check the received data byte (the CTS byte). If the CTS byte is 0x00, the SPI will repeat the polling operation. When the CTS byte reads back non-zero, the SPI will begin transmitting 0x00 to the slave and store the received data in the FIFO. The CTS byte can optionally be stored, by clearing the FLOWMD bit to 0. If FLOWMD is 1, the CTS byte will be ignored.



*First non-zero CTS byte is stored in RX FIFO if FLOWMD = 0

Figure 34.11. CTS Data Polling

When CTSEN is set to 1, pin polling is used, as shown in Figure 34.12. During the polling phase, the SPI will not send clocks to the slave device. Instead, the CTS pin is monitored. When CTS goes high, the SPI will begin transmitting 0x00 to the slave and store the received data in the FIFO. Note that in pin polling mode, the FLOWMD bit will also affect FIFO storage of the first received byte. If set to 1, the first byte of the incoming data will be ignored, and not stored in the receive FIFO.



*First byte is ignored if FLOWMD = 1

Figure 34.12. CTS Pin Polling

34.7.4. Transmit On Request Mode

Transmit-on-request mode is enabled by setting the TXONREQ bit to 1. This is a special flow control mode, available only when the SPI is used as a 4-wire master on a single-master bus. When TXONREQ is set, OPMD, AUTONSS, and CTSEN have no effect on SPI operation.

Transmit-on-request mode uses the CTS pin as a transmit request line. NSS is automatically controlled by the SPI, and the NSSDELAY and NSSCNT fields are used to determine SPI NSS timing as shown in Figure 34.7. Data is transferred in bursts of up to 8 bytes, determined by the NSSCNT field (NSS should be set between 0-7 in this mode). Each transfer burst is initiated by a rising edge on the CTS pin. When a burst is complete, the SPI deasserts NSS and waits for the next rising edge of CTS. If the transmit buffer goes empty during a burst, or if there is no data available in the buffer when a burst is requested, the SPI will stall and set the URI flag. The transfer will continue if additional data becomes available in the transmit FIFO.

Transmit-on-request mode is terminated by disabling the SPI or by setting the ABORT bit. If ABORT is used, the SPI will complete the current data transfer and then disable itself. Figure 34.13 shows the timing for transmit-on-request mode.

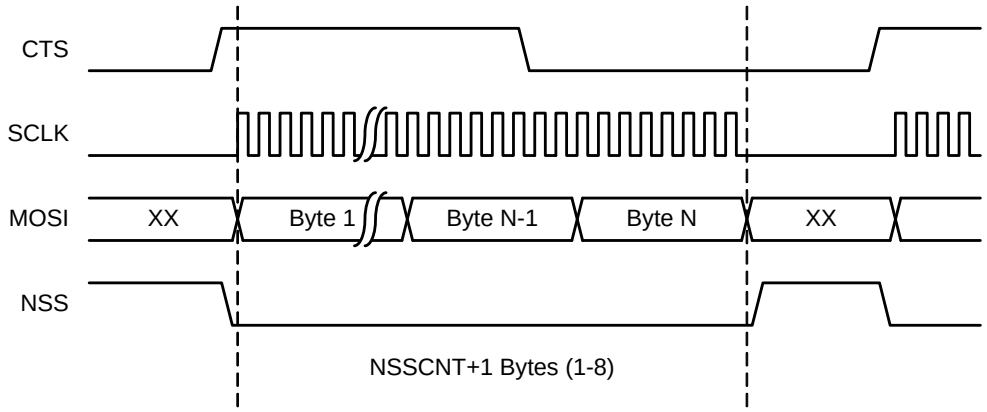


Figure 34.13. Transmit On Request Mode Timing

34.8. Interrupts

The SPI module has several interrupt sources that can generate a SPI interrupt. All sources can be enabled or disabled by a corresponding interrupt enable bit except for the illegal FIFO access interrupts (TFILI and RFILI), which are always enabled.

34.8.1. Transmit Interrupts

The transmit FIFO write request (TFRQI) flag indicates that the FIFO has more room for data. Hardware sets this flag when the number of empty slots in the transmit FIFO is greater than or equal to the number of slots specified in the TFTH field. If DMA operations are enabled, a DMA request will be generated when hardware sets the flag. This flag can also generate an interrupt if the TFRQIEN bit is set to 1 until the number of empty slots in the transmit FIFO level drops below the TFTH setting.

34.8.2. Receive Interrupts

This receive FIFO read request interrupt (RFRQI) flag indicates that the receive FIFO has data available to be read by firmware. Hardware sets this bit when the number of filled slots in the receive FIFO is greater than or equal to the number of slots specified in the RFTH field. A DMA request will be generated if DMA is enabled (DMAEN = 1). If the receive FIFO read request interrupt is enabled (RFRQIEN = 1), an interrupt will also be generated until the number of filled slots in the receive FIFO drops below the RFTH setting.

34.8.3. Other Interrupts

The slave selected interrupt (SLVSELI) flag indicates when the slave select signal (NSS) is active. This flag represents a deglitched version of the NSS pin, rather than the instantaneous pin value. Firmware can read the instantaneous pin value by reading the NSSSTS bit. If the slave selected interrupt enable (SLVSELIEN) bit is also set, an interrupt will be generated. The interrupt will continuously trigger as long as the master asserts the NSS pin.

The shift register empty interrupt (SREI) and underrun interrupt (URI) flags indicate when the hardware has shifted all data out of the shift register and there's no data waiting in the transmit FIFO. The hardware will also generate an interrupt when SREI is set if SREIEN is set to 1 or when URI is set if URIEN is set to 1. These bits must be cleared by firmware.

34.8.4. Error Interrupts

The SPI module also has several error interrupts. The transmit FIFO overrun interrupt (TFORI) flag indicates a transmit data loss condition. This occurs when a write to the DATA register is attempted without sufficient free space in the transmit FIFO. Any new data written to the DATA register will not be placed in the transmit FIFO. The hardware will generate an interrupt if the transmit FIFO overrun interrupt enable (TFORIEN) bit is set. This flag must be cleared by firmware.

When the receive FIFO is full and new data arrives in the shift register, the hardware sets the receive FIFO overrun interrupt (RFORI) flag and ignores the data. This flag can also generate an interrupt if RFORIEN is set to 1. This flag must be cleared by firmware.

The mode fault interrupt (MDFI) flag indicates when a master mode collision occurs on the bus. The hardware sets this flag when NSS is low in multi-master mode (MSTEN = 1 and NSSMD = 1). An interrupt will be generated when MDFI sets, if enabled (MDFIEN= 1). This flag must be cleared by firmware.

The illegal transmit and receive FIFO access interrupts (TFILI and RFILI) are always enabled. These errors can occur when accesses to the DATA register are not right-justified, and an interrupt will be generated if either of these bits is set to 1. These flags must be cleared by firmware.

34.9. Debug Mode

Firmware can set the DBGMD bit to force the SPI module to halt on a debug breakpoint. Clearing the DBGMD bit forces the module to continue operating while the core halts in debug mode.

34.10. Module Reset

The SPI module can be reset by setting the RESET bit in the CONFIG register to 1. This bit resets the SPIEN and MSTEN bits in the CONFIG register, all bits in the CONTROL register, and flushes the receive and transmit FIFOs. The affected bits and fields are inaccessible during the reset process. Firmware should poll the RESET bit until hardware clears it, indicating the reset operation is complete.

34.11. DMA Configuration and Usage

When DMA is enabled (DMAEN = 1), the transmitter will generate a DMA request when the number of empty slots in the transmit FIFO is greater than or equal to the number of slots specified in the TFTH field. The DMA must service the request for data before the last data in the shift register finishes transmission, causing the device to set the underrun interrupt flag (URI). As soon as firmware loads data into the transmit FIFO, the shift register will pop the first data from the transmit FIFO and resume transmissions. Firmware can determine the number of filled slots in the transmit FIFO by reading the transmit FIFO counter (TFCNT) field. In the event of an error, writing a 1 to the transmit FIFO flush (TFIFOFL) bit flushes all the data in the transmit FIFO.

During reception with DMA enabled, the receiver will generate a DMA request when the number of filled slots in the receive FIFO is greater than or equal to the number of slots specified in the RFTH field. If the receive FIFO is full, the DMA must service the DMA request before an overrun occurs. After an overrun condition, the hardware will not write the last data to the receive FIFO and will set the receive FIFO overrun interrupt (RFORI) flag, resulting in an interrupt if the RFORIEN bit is set to 1. At the end of a data transfer, a DMA request will not be generated if the number of filled slots in the receive FIFO is below the number of slots specified in the RFTH field. After clearing the SPIEN bit to disable the module, firmware can pop any remaining data from the receive FIFO using the DATA register until the receive FIFO counter (RFCNT) field reaches zero. The receive FIFO can also be emptied using the receive FIFO flush (RFIFOFL) bit.

Figure 34.14 shows the SPI DMA configuration.

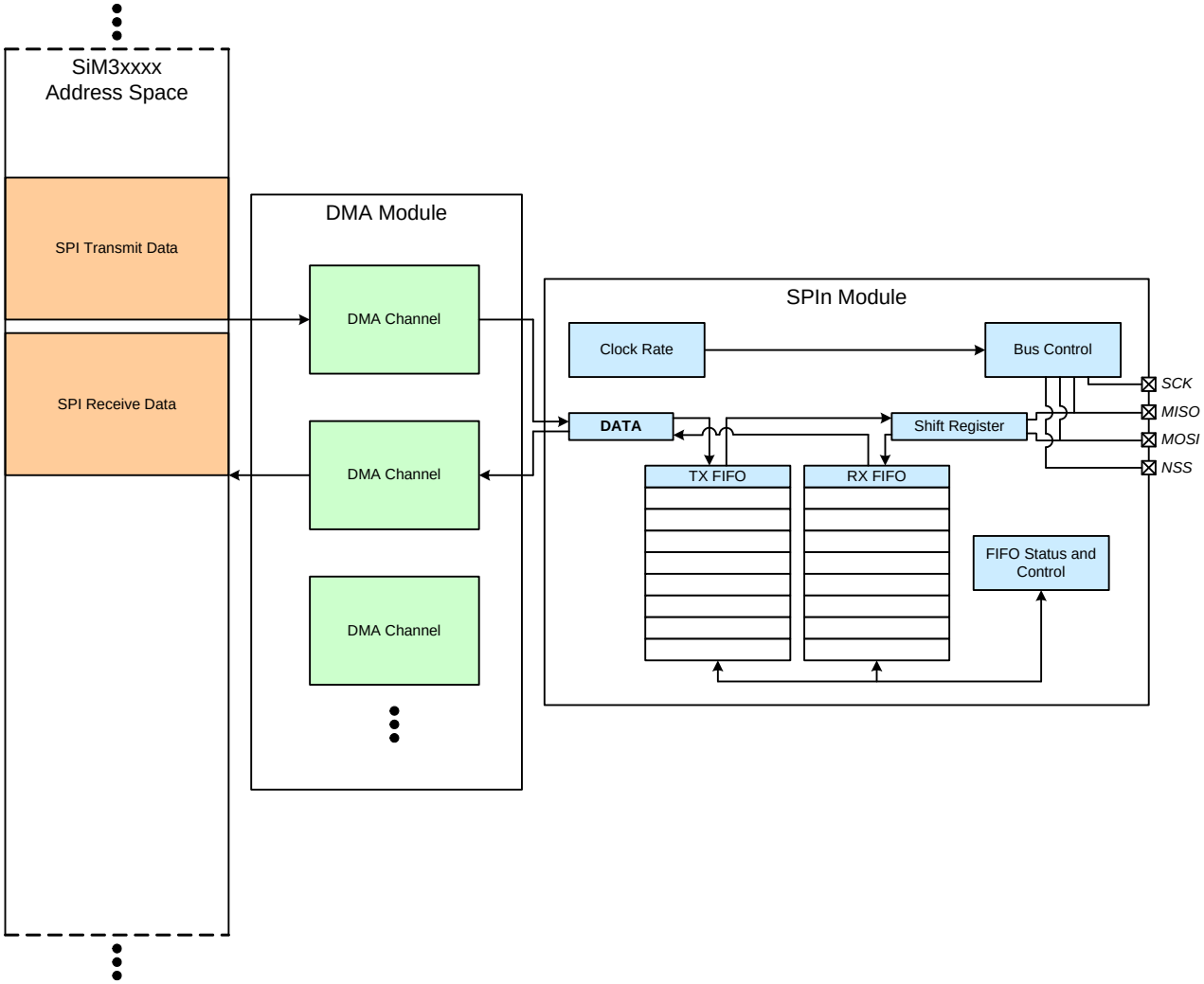


Figure 34.14. SPI DMA Configuration

34.12. SPI0 and SPI1 Registers

This section contains the detailed register descriptions for SPI0 and SPI1 registers.

Register 34.1. SPIn_DATA: Input/Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Addresses

SPI0_DATA = 0x4000_4000
SPI1_DATA = 0x4000_5000

Table 34.2. SPIn_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**Input/Output Data.** The DATA register is used to write to the transmit buffer and read from the receive buffer. Data can be in byte, half-word, or word format and must be right-justified. For every byte of data written, the TFCNT counter will increase. For every byte read, the RFCNT field will decrease.

Notes:
1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

Register 34.2. SPIn_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved							DBGMD	TFCNT				RFCNT			
Type	R							RW	R				R			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	BUSYF	NSSSTS	Reserved				TFILI	RFILI	SREI	URI	MDFI	SLVSELI	TFORI	TFRQI	RFORI	RFRQI
Type	R	R	R				RW	RW	R	RW	RW	R	RW	R	RW	R
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0	0

Register ALL Access Addresses

SPI0_CONTROL = 0x4000_4010

SPI1_CONTROL = 0x4000_5010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 34.3. SPIn_CONTROL Register Bit Descriptions

Bit	Name	Function
31:25	Reserved	Must write reset value.
24	DBGMD	**SPI Debug Mode.** 0: The SPI module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the SPI module to halt.
23:20	TFCNT	**Transmit FIFO Counter.** Indicates the number of bytes in the transmit FIFO.
19:16	RFCNT	**Receive FIFO Counter.** Indicates the number of bytes in the receive FIFO.
15	BUSYF	**SPI Busy.** 0: The SPI is not busy and a transfer is not in progress. 1: The SPI is currently busy and a transfer is in progress.
14	NSSSTS	**NSS Instantaneous Pin Status.** This represents the instantaneous logic level at the NSS pin. 0: NSS is currently a logic low. 1: NSS is currently a logic high.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 34.3. SPIn_CONTROL Register Bit Descriptions

Bit	Name	Function
13:10	Reserved	Must write reset value.
9	TFILI	**Illegal Transmit FIFO Access Interrupt Flag.** This bit indicates that an illegal write of the transmit FIFO has occurred. An interrupt will be generated if this bit is set to 1. This bit must be cleared by firmware.
8	RFILI	**Illegal Receive FIFO Access Interrupt Flag.** This bit indicates that an illegal read of the receive FIFO has occurred. An interrupt will be generated if this bit is set to 1. This bit must be cleared by firmware.
7	SREI	**Shift Register Empty Interrupt Flag.** Indicates that all the data has been transferred out of the shift register and there is no data waiting in the TX FIFO. If SREIEN is enabled, an interrupt will be generated. 0: There is data still present in the transmit FIFO or shift register. 1: All data has been transferred out of the shift register and there is no data waiting in the transmit FIFO.
6	URI	**Underrun Interrupt Flag.** This bit is set to 1 to indicate the end of a data transfer when the transmit FIFO and the shift register are empty. If URIEN is enabled, an interrupt will be generated. This bit must be cleared by firmware.
5	MDFI	**Mode Fault Interrupt Flag.** This bit is set to logic 1 by hardware when a master mode collision is detected (NSS is low, MSTEN = 1, and NSSMD [1:0] = 01b). An interrupt will be generated if MDFIEN is enabled. This bit must be cleared by firmware.
4	SLVSELI	**Slave Selected Interrupt Flag.** Indicates when the slave select signal (NSS) is active. This bit does not represent the instantaneous pin value, but is a deglitched version of the pin. An interrupt will be generated if SLVSELIEN is set to 1. 0: The slave select signal (NSS) is not active. 1: The slave select signal (NSS) is active.
3	TFORI	**Transmit FIFO Overrun Interrupt Flag.** Indicates that a transmit FIFO overrun has occurred and the written data will not be placed in the FIFO. If TFORIEN is enabled, an interrupt will be generated. This bit must be cleared by firmware.
2	TFRQI	**Transmit FIFO Write Request Interrupt Flag.** This flag indicates that the TX FIFO is at or below the number of bytes defined by the TFTH field. If DMA is enabled, a DMA request will be generated. If TFRQIEN is set to 1, an interrupt will be generated until the FIFO level fills above TFTH. 0: The TX FIFO has fewer empty slots than the level defined by TFTH. 1: The TX FIFO has at least as many empty slots as the level defined by TFTH.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 34.3. SPIn_CONTROL Register Bit Descriptions

Bit	Name	Function
1	RFORI	**Receive FIFO Overrun Interrupt Flag.** This flag Indicates that a receive FIFO overrun has occurred and the new data will not be placed in the FIFO. If RFORIEN is enabled, an interrupt will be generated. This bit must be cleared by firmware.
0	RFRQI	**Receive FIFO Read Request Interrupt Flag.** This flag indicates that the RX FIFO is at or above the number of bytes defined by the RFTH field. If DMA is enabled, a DMA request will be generated. An interrupt will be generated if RFRQIEN is set to 1 until the FIFO level drops below RFTH. 0: The RX FIFO has fewer bytes than the level defined by RFTH. 1: The RX FIFO has equal or more bytes than the level defined by RFTH.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 34.3. SPIn_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RESET	TFIFOFL	RFIFOFL	Reserved				DMAEN	DSIZE				TFTH		RFTH	
Type	RW	RW	RW	R				RW	RW				RW		RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	NSSMD		DDIRSEL	NSSPOL	CLKPHA	CLKPOL	MSTEN	SPIEN	SREIEN	URIEN	MDFIEN	SLVSELIEN	TFORIEN	TFRQIEN	RFORIEN	RFRQIEN
Type	RW		RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

SPI0_CONFIG = 0x4000_4020
SPI1_CONFIG = 0x4000_5020
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 34.4. SPIn_CONFIG Register Bit Descriptions

Bit	Name	Function
31	RESET	**Module Soft Reset.** Setting this bit to 1 resets the SPIEN and MSTEN bits in the CONFIG register, all bits in the CONTROL register, and flushes the RX and TX FIFOs. This bit is cleared by hardware when the operation completes. 0: SPI module is not in soft reset. 1: SPI module is in soft reset and some of the module bits cannot be accessed until this bit is cleared to 0 by hardware.
30	TFIFOFL	**Transmit FIFO Flush.** Setting this bit to 1 flushes the transmit FIFO of all data. This bit is cleared by hardware when the operation completes.
29	RFIFOFL	**Receive FIFO Flush.** Setting this bit to 1 flushes the receive FIFO of all data. This bit is cleared by hardware when the operation completes.
28:25	Reserved	Must write reset value.
24	DMAEN	**DMA Enable.** 0: Disable DMA requests. 1: Enable DMA requests according to the TFRQI and RFRQI flags.

Table 34.4. SPIn_CONFIG Register Bit Descriptions

Bit	Name	Function
23:20	DSIZE	**Data Size.** This field specifies the length of a data transfer. The SPI supports data transfers of any length between 1 and 16 bits. The data transfer size is equal to DSIZE + 1.
19:18	TFTH	**Transmit FIFO Threshold.** This field sets the trip point for transmit FIFO requests. 00: A DMA / TFRQ request asserts when ≥ 1 FIFO slot is empty. 01: A DMA / TFRQ request asserts when ≥ 2 FIFO slots are empty. 10: A DMA / TFRQ request asserts when ≥ 4 FIFO slots are empty. 11: A DMA / TFRQ request asserts when all FIFO slots are empty.
17:16	RFTH	**Receive FIFO Threshold.** This field sets the trip point for receive FIFO requests. 00: A DMA / RFRQ request asserts when ≥ 1 FIFO slot is filled. 01: A DMA / RFRQ request asserts when ≥ 2 FIFO slots are filled. 10: A DMA / RFRQ request asserts when ≥ 4 FIFO slots are filled. 11: A DMA / RFRQ request asserts when all FIFO slots are filled.
15:14	NSSMD	**Slave Select Mode.** This bit selects the behavior of the NSS pin. 00: 3-wire Slave or 3-wire Master. 01: 4-wire slave (NSS input). This setting can also be used for multi-master configurations. 10: 4-wire master with NSS low (NSS output). 11: 4-wire master with NSS high (NSS output).
13	DDIRSEL	**Data Direction Select.** 0: Data will be shifted MSB first. 1: Data will be shifted LSB first.
12	NSSPOL	**Slave Select Polarity Select.** 0: NSS is active low. 1: NSS is active high.
11	CLKPHA	**SPI Clock Phase.** 0: The first edge of SCK is the sample edge (center of data bit). 1: The first edge of SCK is the shift edge (edge of data bit).
10	CLKPOL	**SPI Clock Polarity.** 0: The SCK line is low in the idle state. 1: The SCK line is high in the idle state.
9	MSTEN	**Master Mode Enable.** 0: Operate in slave mode. 1: Operate in master mode.
8	SPIEN	**SPI Enable.** 0: Disable the SPI. 1: Enable the SPI.

Table 34.4. SPIn_CONFIG Register Bit Descriptions

Bit	Name	Function
7	SREIEN	**Shift Register Empty Interrupt Enable.** 0: Disable the shift register empty interrupt. 1: Enable the shift register empty interrupt.
6	URIEN	**Underrun Interrupt Enable.** 0: Disable the underrun interrupt. 1: Enable the underrun interrupt.
5	MDFIEN	**Mode Fault Interrupt Enable.** 0: Disable the mode fault interrupt. 1: Enable the mode fault interrupt.
4	SLVSELIEN	**Slave Selected Interrupt Enable.** 0: Disable the slave select interrupt. 1: Enable the slave select interrupt.
3	TFORIEN	**Transmit FIFO Overrun Interrupt Enable.** 0: Disable the transmit FIFO overrun interrupt. 1: Enable the transmit FIFO overrun interrupt.
2	TFRQIEN	**Transmit FIFO Write Request Interrupt Enable.** 0: Disable the transmit FIFO data request interrupt. 1: Enable the transmit FIFO data request interrupt.
1	RFORIEN	**Receive FIFO Overrun Interrupt Enable.** 0: Disable the receive FIFO overrun interrupt. 1: Enable the receive FIFO overrun interrupt.
0	RFRQIEN	**Receive FIFO Read Request Interrupt Enable.** 0: Disable the receive FIFO request interrupt. 1: Enable the receive FIFO request interrupt.

Register 34.4. SPIn_CLKRATE: Module Clock Rate Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	CLKDIV															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

SPI0_CLKRATE = 0x4000_4030

SPI1_CLKRATE = 0x4000_5030

Table 34.5. SPIn_CLKRATE Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:0	CLKDIV	**Clock Divider.** This field sets the frequency of the SPI clock output in master mode, according to the equation: $F_{SCK} = \frac{F_{APB}}{2 \times (CLKDIV + 1)}$

Register 34.5. SPIn_FSTATUS: FIFO Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	TFWPTR				TFRPTR				RFWPTR				RFRPTR			
Type	R				R				R				R			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses
SPI0_FSTATUS = 0x4000_4040
SPI1_FSTATUS = 0x4000_5040

Table 34.6. SPIn_FSTATUS Register Bit Descriptions

Bit	Name	Function
31:16	Reserved	Must write reset value.
15:12	TFWPTR	**Transmit FIFO Write Pointer.** This field indicates the next slot firmware will access on a transmit FIFO write.
11:8	TFRPTR	**Transmit FIFO Read Pointer.** This field indicates the next slot the SPI hardware will read from the transmit FIFO.
7:4	RFWPTR	**Receive FIFO Write Pointer.** This field indicates the next slot the SPI hardware will write in the receive FIFO.
3:0	RFRPTR	**Receive FIFO Read Pointer.** This field indicates the next slot firmware will access on a receive FIFO read.

Register 34.6. SPIn_CONFIGMD: Mode Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved					TFRCNT			NSSDELAY							
Type	R					RW			RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	NSSCNT								Reserved	TXONREQ	ABORT	FLOWMD	CTSEN	AUTONSS	OPMD	
Type	RW								R	RW	RW	RW	RW	RW	RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

SPI0_CONFIGMD = 0x4000_4050
SPI1_CONFIGMD = 0x4000_5050
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 34.7. SPIn_CONFIGMD Register Bit Descriptions

Bit	Name	Function
31:27	Reserved	Must write reset value.
26:24	TFRCNT	**Transfer Count.** This field sets the maximum number of data that are written to or read from the TX/RX FIFO per write/read operation. 000: Automatic. The request type determines the number of bytes to write/read. 001: A single byte is written/read per request. 010: Up to two bytes can be written/read per request. 011: Up to three bytes can be written/read per request. 100: Up to four bytes can be written/read per request. 101-111: Reserved.
23:16	NSSDELAY	**NSS Delay.** When AUTONSS is equal to 1, this field determines the delay time between NSS assertions. The delay time will be equivalent to (NSSDELAY + 1) / 2 SPI clock periods.
15:8	NSSCNT	**NSS Data Count.** Determines the number of bytes transferred per NSS assertion when AUTONSS or TXONREQ is enabled. The number of data bytes per NSS assertion is equal to NSSCNT + 1. When used with TXONREQ, the maximum value for NSSCNT is 7 (8 bytes per transfer).
7	Reserved	Must write reset value.

Table 34.7. SPIn_CONFIGMD Register Bit Descriptions

Bit	Name	Function
6	TXONREQ	**Transmit On Request.** This bit is valid in 4-wire single-master mode only. When set data will be transmitted in bursts on each rising edge of the CTS signal. The CTSEN, AUTONSS and OPMD bits are all ignored when TXONREQ is set.
5	ABORT	**Software Abort.** When firmware sets this bit the SPI will complete the current data transfer and then disable itself. Hardware clears this bit when the software abort is complete.
4	FLOWMD	**Flow Control Mode.** Determines whether CTS data is stored when operating in flow control mode. 0: Store CTS byte (see chapter text for details). 1: Disregard CTS byte.
3	CTSEN	**CTS Flow Control Enable.** Enables CTS flow control.
2	AUTONSS	**Auto NSS Mode.** Enables automatic NSS toggling when in 4-wire master mode (NSSMD[1] = 1).
1:0	OPMD	**Operation Mode.** 00: Full-duplex (normal) mode. 01: Receive-only mode. 10: Transmit only mode. 11: Flow control mode.

34.13. SPIn Register Memory Map

Table 34.8. SPIn Memory Map

Register Name	SPIn_CLKRATE	SPIn_CONFIG	SPIn_CONTROL	SPIn_DATA
ALL Offset	0x30	0x20	0x10	0x0
Access Methods	ALL	ALL | SET | CLR	ALL | SET | CLR	ALL
Bit 31	Reserved	RESET	Reserved	DATA
Bit 30		TFIFOFL		
Bit 29		RFIFOFL		
Bit 28		Reserved		
Bit 27				
Bit 26				
Bit 25				
Bit 24		DMAEN	DBGMD	
Bit 23		DSIZE	TFCNT	
Bit 22				
Bit 21				
Bit 20				
Bit 19		TFTH	RFCNT	
Bit 18				
Bit 17		RFTH		
Bit 16				
Bit 15	CLKDIV	NSSMD	BUSYF	
Bit 14			NSSSTS	
Bit 13		DDIRSEL	Reserved	
Bit 12		NSSPOL		
Bit 11		CLKPHA		
Bit 10		CLKPOL		
Bit 9		MSTEN	TFILI	
Bit 8		SPIEN	RFILI	
Bit 7		SREIEN	SREI	
Bit 6		URIEN	URI	
Bit 5		MDFIEN	MDFI	
Bit 4		SLVSELIEN	SLVSELI	
Bit 3		TFORIEN	TFORI	
Bit 2		TFRQIEN	TFRQI	
Bit 1		RFORIEN	RFORI	
Bit 0		RFRQIEN	RFRQI	

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: SPI0 = 0x4000_4000, SPI1 = 0x4000_5000

Table 34.8. SPIn Memory Map

Register Name	ALL Offset	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SPIn_FSTATUS	0x40	ALL	Reserved																TFWPTR				TFRPTR				RFWPTR				RFRPTR			
SPIn_CONFIGMD	0x50	ALL | SET | CLR	Reserved																NSSCNT								Reserved	TXONREQ	ABORT	FLOWMD	CTSEN	AUTONSS	OPMD	

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: SPI0 = 0x4000_4000, SPI1 = 0x4000_5000

35. Timers (TIMER0, TIMER1 and TIMER2)

This section describes the TIMER module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the TIMER block, which is used by TIMER0, TIMER1 and TIMER2 on all device families covered in this document.

35.1. Timer Features

Each timer module (TIMER) is independent, and includes the following features:

- Operation as a single 32-bit or two independent 16-bit timers.
- Clocking options include the APB clock, the APB clock scaled using an 8-bit clock divider, the external oscillator, or falling edges on an external input pin (synchronized to the APB clock).
- Auto-reload functionality in both 32-bit and 16-bit modes.
- Up/Down count capability, controlled by an external input pin.
- Rising and falling edge capture modes on an external pin (TIMER0 and TIMER1).
- Capture PVTOSC rising and falling edges (TIMER2).
- Low or high pulse capture modes.
- Frequency and duty cycle capture modes.
- One shot mode, triggered from EPCA Sync signal.
- Square wave output mode, which is capable of toggling an external pin at a given rate with 50% duty cycle.
- 32-bit or 16-bit pulse-width modulation modes.

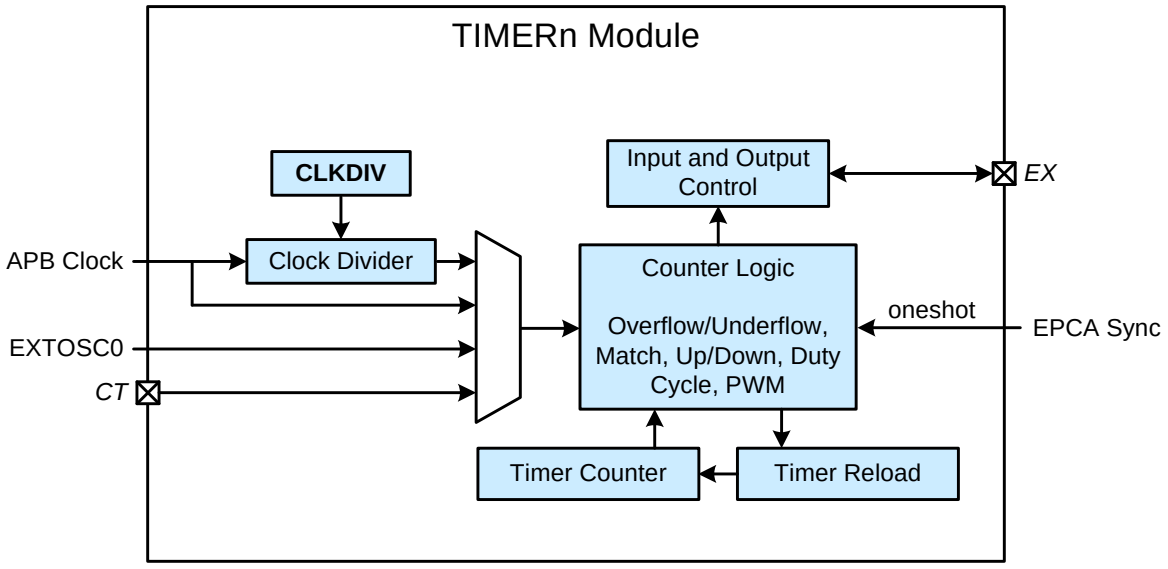


Figure 35.1. Timer Block Diagram

35.2. Clocking

The timer module can be clocked from several internal and external sources, selected by the HCLK and LCLK bits. The SPLITEN bit allows the two halves of the 32-bit timer to be split and clocked as individual 16-bit timers. When operating as a single 32-bit timer (SPLITEN = 0), HCLK controls the clock to the entire timer. When configured for split mode (SPLITEN = 1), HCLK controls the clock to the high-side 16-bit timer, and LCLK controls the clock to the low-side 16-bit timer.

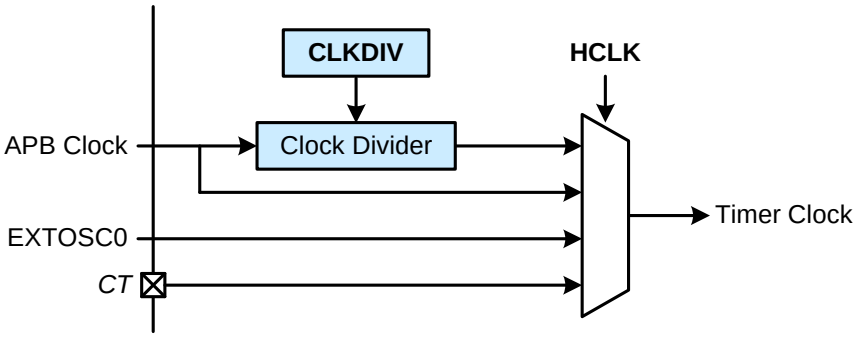


Figure 35.2. Clock Source Selection (SPLITEN = 0)

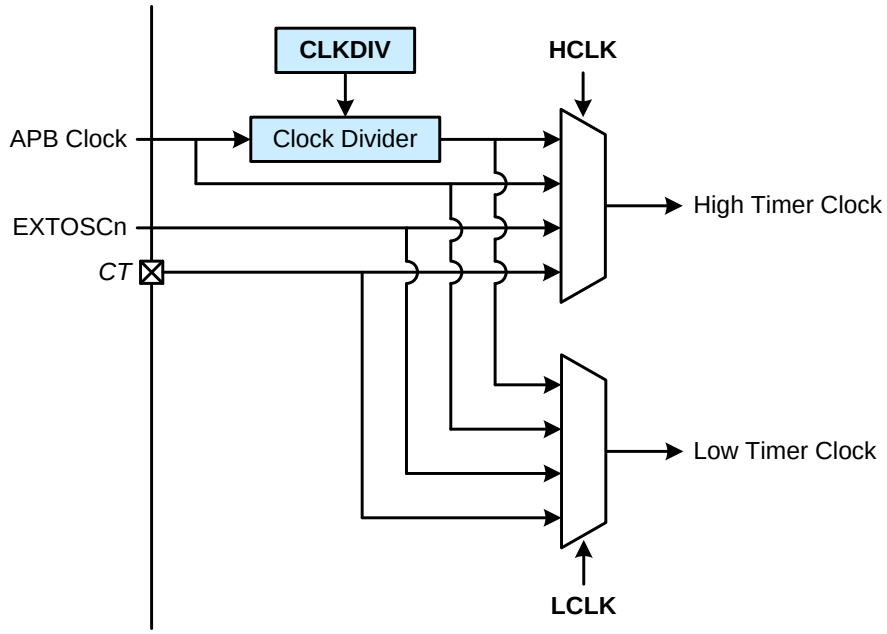


Figure 35.3. Clock Source Selection (SPLITEN = 1)

35.2.1. APB Clock

When the APB clock signal is selected as the timer clock source, the timer will clock from the APB clock source defined by the Clock Control module.

35.2.2. External Clock (EXTOSCn)

When the external clock is selected as the timer clock source, the timer will clock from the external clock source (EXTOSCn), regardless of the clock selection of the core. In this mode the external clock source is synchronized to the selected APB clock. In order to guarantee that the external clock transitions are recognized by the device, the external clock signal must be high or low for at least one APB clock. This limits the maximum frequency of an external clock in this mode to one-half the APB clock.

35.2.3. Clock Divider

The timer module includes an 8-bit configurable clock divider, allowing the timer to be clocked by a divided version of the APB clock. The clock divider consists of an 8-bit up counter and reload register and runs only when selected as the timer clock source. The counter value is incremented on every tick of the APB clock. It can be directly read and written through the CLKDIVCT field, and the reload value is stored in the CLKDIVRL field. On an overflow from 0xFF, the CLKDIVCT counter is automatically reloaded with the value in CLKDIVRL. The resulting timer clock rate is determined by Equation 35.1.

$$F_{TIMER} = \frac{F_{APB}}{(256 - CLKDIVRL)}$$

Equation 35.1. Clock Divider Output Clock Rate

35.2.4. External Pin (CT)

When the CT pin is selected as the timer clock source, the timer is incremented on falling edges of the pin. The CT pin is synchronized to the selected APB clock in this mode. In order to guarantee that the CT pin transitions are recognized by the device, CT must be at its new logic state for at least two APB clock cycles. Thus, the maximum frequency at which CT can toggle is one fourth the APB clock speed.

35.3. Configuring Timer Interrupts

There are two interrupt sources each for the high and low 16-bit words in the timer module. Each interrupt source can be individually enabled to generate a timer interrupt. The available interrupt flags are HOVFI, LOVFI, HEXI, and LEXI. The overflow flags (HOVFI and LOVFI) are set to 1 any time a positive overflow occurs (an increment when the timer is all 1's). The overflow interrupt enable bits (HOVFIEN and LOVFIEN) enable the corresponding overflow flags to be recognized by the interrupt controller. The extra flags (HEXI and LEXI) are set to 1 under conditions defined by the selected timer mode. The corresponding enable bits (HEXIEN and LEXIEN) enable the extra flags to be recognized by the interrupt controller. Refer to the timer mode descriptions in Section 35.5 for specifics on the overflow and extra flag triggers for each mode.

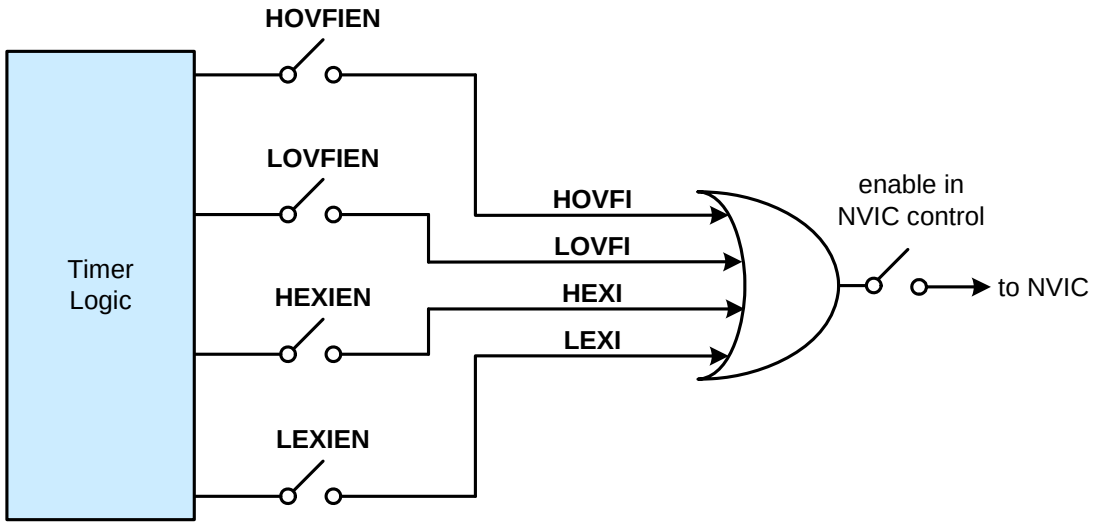


Figure 35.4. Timer Interrupt Configuration

35.4. Timer Synchronization

Each 16-bit timer includes a synchronization mechanism that allows all timers on the device to start and stop simultaneously. This synchronization is controlled using the master enable bits HMSTREN (high timer) and LMSTREN (low timer) and the master run control bit (MSTRUN) in the master TIMER module (TIMER0). When operating in 32-bit mode, HMSTREN controls the full timer. The MSTRUN bit affects all timer modules (0 through m) and is defined in the CONFIG register. If the master enable bit (HMSTREN, LMSTREN) is set for a timer, both MSTRUN in the master timer (TIMER0) and HRUN must be set to 1 for the timer to run.

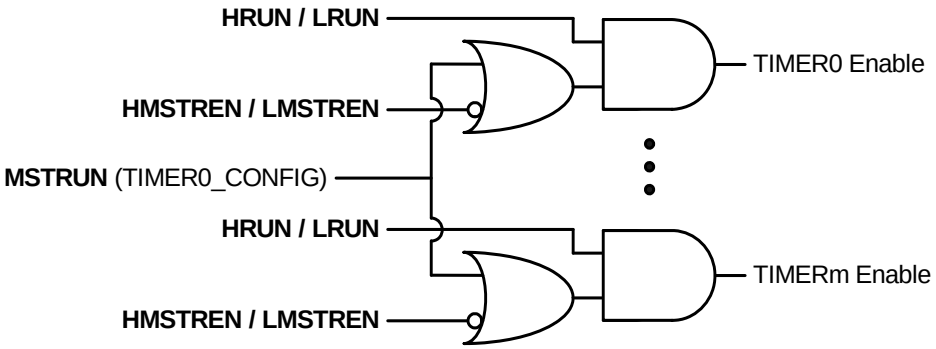


Figure 35.5. Timer Synchronization Block Diagram

35.5. Timer Modes

The timer mode selection is accomplished using the mode select bits in conjunction with the SPLITEN bit. All timer modes are available when operating as a 32-bit timer (SPLITEN = 0). Modes that use the EX pin as an input can be used with either the high or low timer in 16-bit split mode (SPLITEN = 1). In these modes, the two 16-bit timers share the EX input pin. Modes using the pin as an output are only available to the high timer in 16-bit split mode. Table 35.1 shows the different timer modes with their availability and external pin functions.

Table 35.1. Timer Mode Selection and Availability

Timer Mode	Mode Control Field	32-bit Mode (SPLITEN = 0)	16-bit Mode (SPLITEN = 1)	EX Pin
Auto-Reload	0000b	Available	Available for High and Low	Not Used
Up/Down	0001b	Available	Available for High and Low	Direction Input
Falling Edge Capture	0010b	Available	Available for High and Low	Capture Input
Rising Edge Capture	0011b	Available	Available for High and Low	Capture Input
Low Pulse Capture	0100b	Available	Available for High and Low	Capture Input
High Pulse Capture	0101b	Available	Available for High and Low	Capture Input
Duty Cycle Capture	0110b	Available	Available for High and Low	Capture Input
One Shot	0111b	Available	Available for High and Low	Not Used
Square Wave	1000b	Available	Available for High only	Square Wave Output
Pulse Width Modulation	1001b	Available	Available for High only	PWM Output

When operating in a 32-bit mode, the entire 32-bit COUNT register is used to store the current timer value. The 32-bit CAPTURE register function is defined by the mode in which the timer is operating. For split 16-bit mode, the corresponding high and low 16-bits of these registers are used independently by the high and low timer.

35.5.1. EX Input Synchronization

When used as an input in a timer mode, the EX signal is synchronized with the APB clock. The timer response delay to EX signal changes is 3-4 APB clocks, and the EX signal must remain high or low for two APB rising edges to be recognized by the timer. The timer event (capture, start or stop, or count direction change) associated with the EX signal will occur on the 4th rising APB edge after the EX signal change, as shown in Figure 35.6.

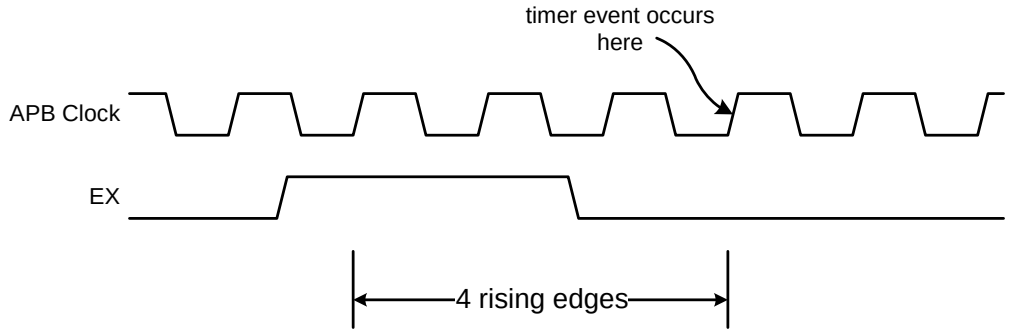


Figure 35.6. EX Signal Synchronization Timing

35.5.2. Auto-Reload Mode

In auto-reload mode, the timer counts up to all 1's. When an overflow occurs, the count register reloads from the value stored in the auto-reload register, and a timer overflow interrupt is generated.

If operating in 16-bit mode, Equation 35.2 describes the high timer's overflow rate. The low timer's overflow rate is based on the LCCR value instead of HCCR.

$$\text{Overflow Rate} = \frac{F_{TIMER}}{65536 - HCCR}$$

Equation 35.2. 16-bit Auto-Reload Overflow Rate

Equation 35.3 describes the timer's overflow rate if operating in 32-bit mode.

$$\text{Overflow Rate} = \frac{F_{TIMER}}{4294967296 - CAPTURE}$$

Equation 35.3. 32-bit Auto-Reload Overflow Rate

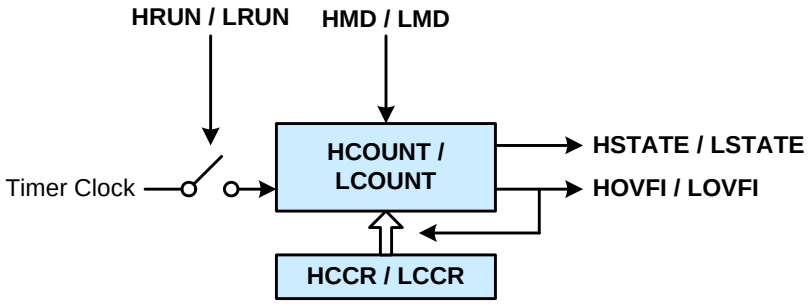


Figure 35.7. Auto-Reload Mode Block Diagram

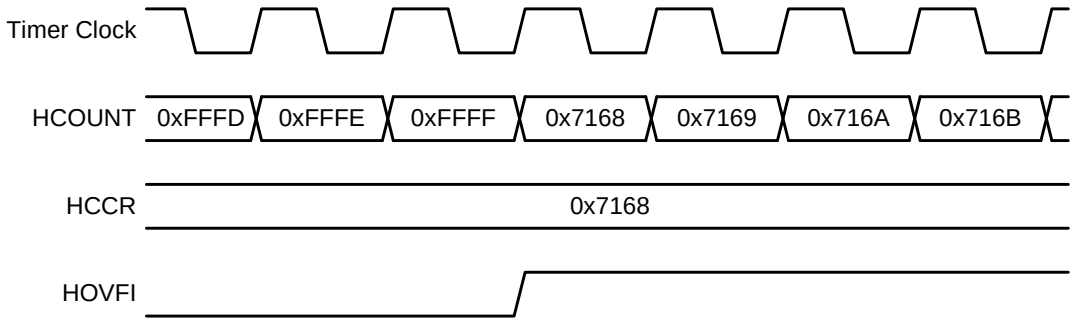


Figure 35.8. 16-bit Auto-Reload Mode Timing Diagram

35.5.3. Up/Down Mode

In up/down mode, the timer counts up or down, depending on the state of the external EX pin (high = count up, low = count down). If an overflow occurs, the count register is reloaded from the value stored in the auto-reload register, and a timer overflow is generated. If the timer counts down past the auto-reload value, the count register is reloaded with all 1's, and a timer underflow is generated (HEXI or LEXI set to 1).

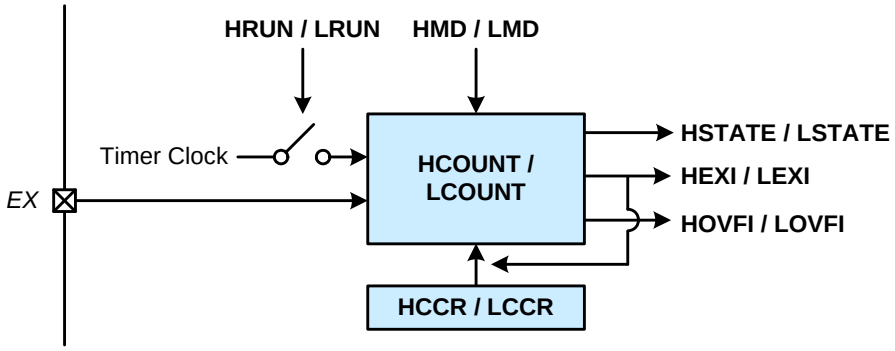
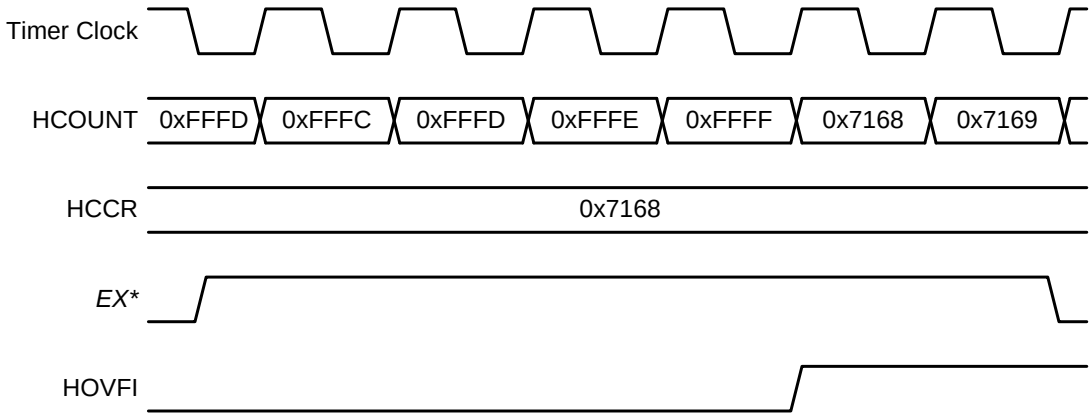


Figure 35.9. Up/Down Mode Block Diagram



*Note: The timer response delay to EX input signal changes is 3-4 APB clock cycles.

Figure 35.10. 16-bit Up/Down Mode Timing Diagram

35.5.4. Edge Capture Mode (Rising or Falling)

When operated in either falling or rising edge capture mode, the timer module counts up to all 1's. The overflow flag is set when an overflow occurs, and the timer reloads to all 0's. If the selected edge (rising or falling) occurs on the EX pin, the timer value is captured to the capture/compare/reload register and the HEXI or LEXI flag is set.

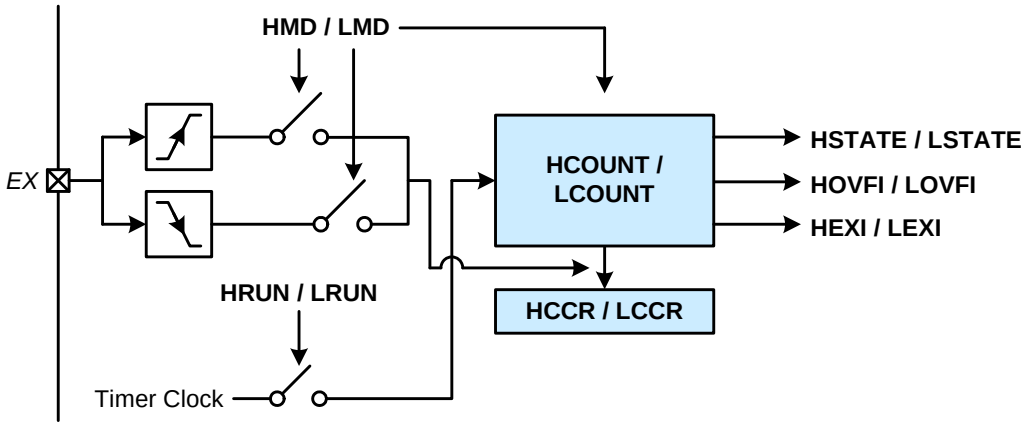
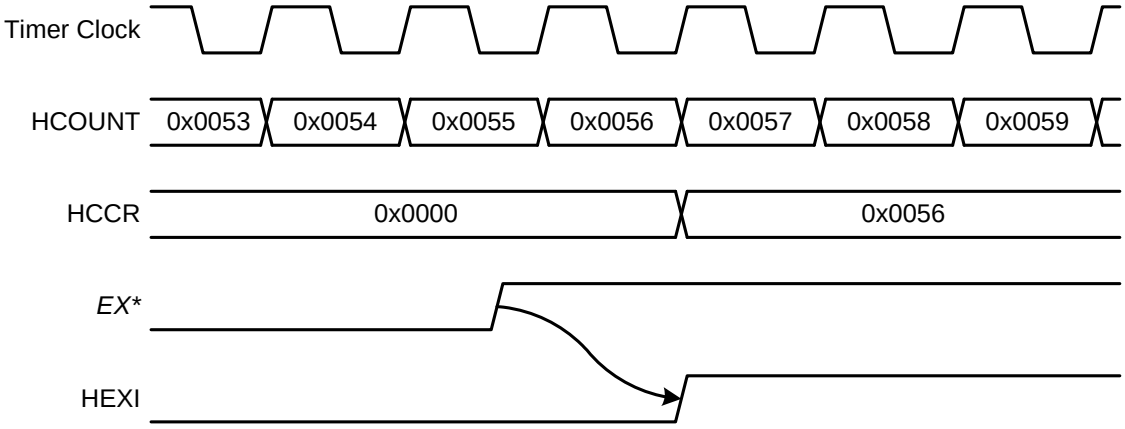


Figure 35.11. Edge Capture Mode Block Diagram



*Note: The timer response delay to EX input signal changes is 3-4 APB clock cycles.

Figure 35.12. 16-bit Edge Capture Mode Timing Diagram (Rising Edge Shown)

35.5.5. Pulse Capture Mode (High or Low)

When operating in low or high pulse capture modes, the timer looks for a sequence of two edges that define a pulse on the external EX pin. In low pulse capture mode, the first edge is the first falling edge the pin sees, while the second edge is the rising edge that follows. In high pulse capture mode, the first edge is a rising edge and the second edge is a falling edge.

The timer is configured to first wait for the first edge, and capture the current timer value to the capture/compare/reload register. The HSTATE or LSTATE bit is set high when this first edge occurs. After the capture occurs, the timer continues to run until the second edge occurs. The timer will halt on the second edge, effectively capturing the second edge position in the main timer counter. On the second edge, the HEXI or LEXI flag is set. The difference between the two captured edges can be used to determine the pulse width.

Note: The timer automatically clears its HRUN or LRUN bit when the second edge is detected.

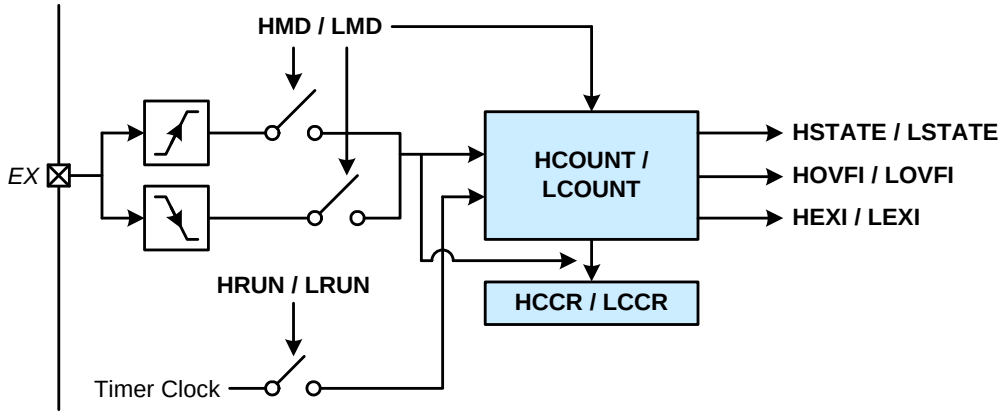
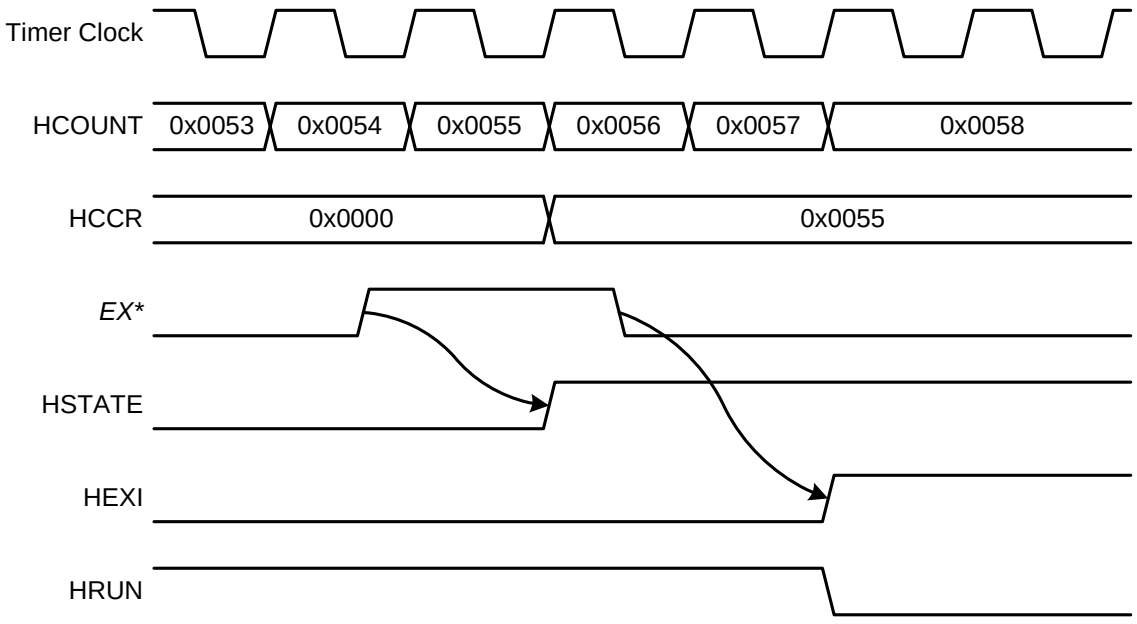


Figure 35.13. Pulse Capture Mode Block Diagram



*Note: The timer response delay to EX input signal changes is 3-4 APB clock cycles.

Figure 35.14. Pulse Capture Mode Timing Diagram (High Pulse Shown)

35.5.6. Duty Cycle Capture Mode

When operating in duty cycle capture mode, the timer initially sits idle (not clocking with HRUN and LRUN cleared to 0) and waits for a sequence of three edges on the EX pin. The timer begins running (by setting its HRUN or LRUN bit) on the first rising edge. The second edge, which is a falling edge, triggers a capture of the current timer value into the capture/compare/reload register. When the capture occurs, the HSTATE or LSTATE bit is set high. The timer halts by clearing its HRUN or LRUN bit on the third edge (another rising edge), effectively capturing the third edge position in the main timer counter. On the third edge, the HEXI or LEXI flag is also set.

The difference between the initial timer value and the final timer value is equivalent to the duration between two rising edges, and can be used to determine the frequency or period of the sampled signal. The difference between the capture value and the initial timer value is the high time of the signal, while the difference between the final timer value and the capture value is the low time of the signal. These can be used to determine the duty cycle of the sampled signal.

Note: The timer automatically sets and clears its HRUN or LRUN bits when edges occur on the EX pin.

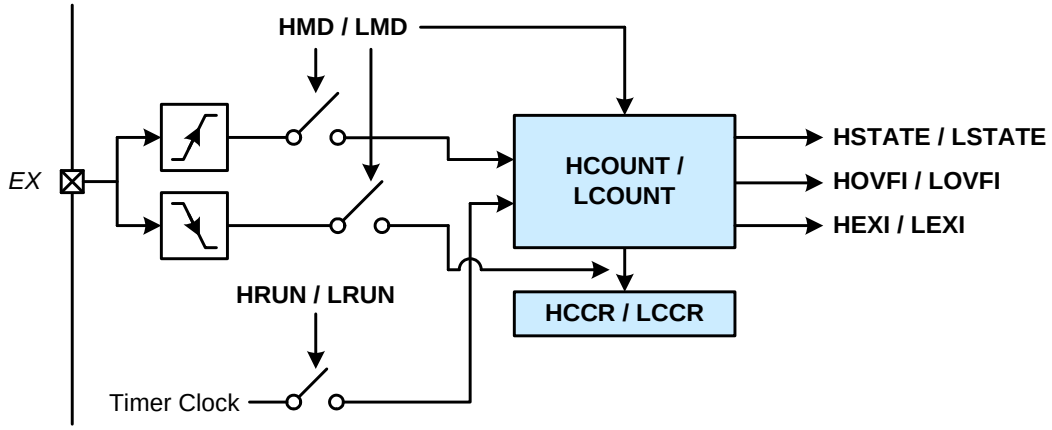
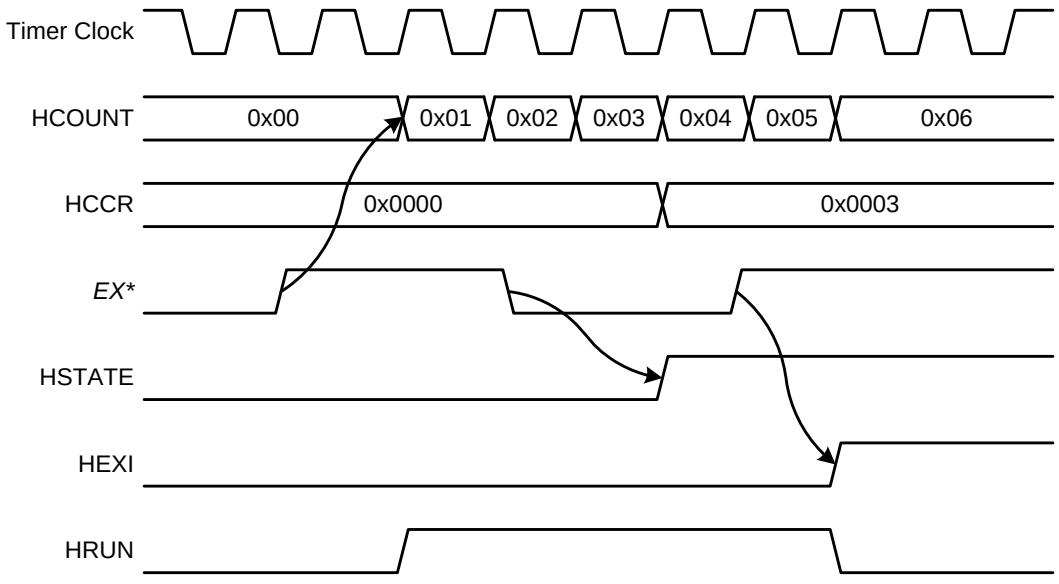


Figure 35.15. Duty Cycle Capture Mode Block Diagram



*Note: The timer response delay to EX input signal changes is 3-4 APB clock cycles.

Figure 35.16. Duty Cycle Capture Mode Timing Diagram

35.5.7. One Shot Mode

In one shot mode, the timer is initially idle (not clocking with HRUN and LRUN cleared to 0) and waiting for a pulse on its internal oneshot input. When a pulse is detected on the oneshot input, the timer begins running (by setting its HRUN or LRUN bit), and operates as though it were in auto-reload mode. The timer counts up to all 1's, and overflows. When the overflow occurs, the count register reloads from the value stored in the auto-reload register, and will immediately halt (by clearing HRUN or LRUN to 0). If enabled, a timer overflow interrupt is generated. The timer will remain idle until the next oneshot pulse occurs.

In the SiM3L1xx device family, all of the timer oneshot signals are tied to the EPCA Sync output. This allows the timers to implement a delayed trigger for different peripherals off the EPCA Sync event.

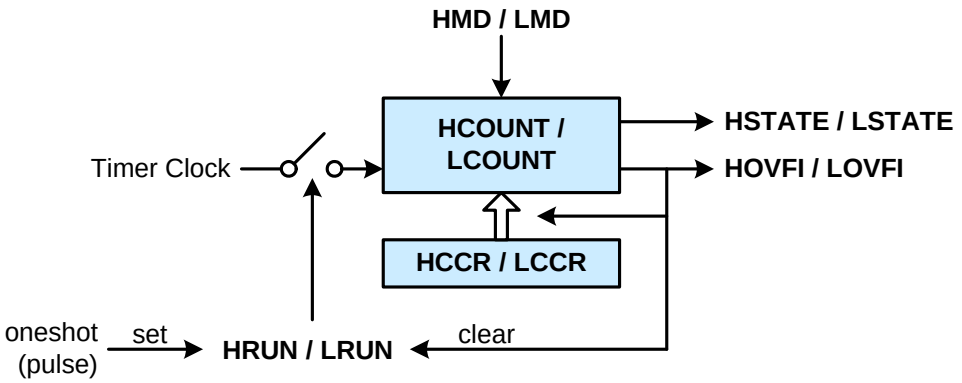


Figure 35.17. One Shot Mode Block Diagram

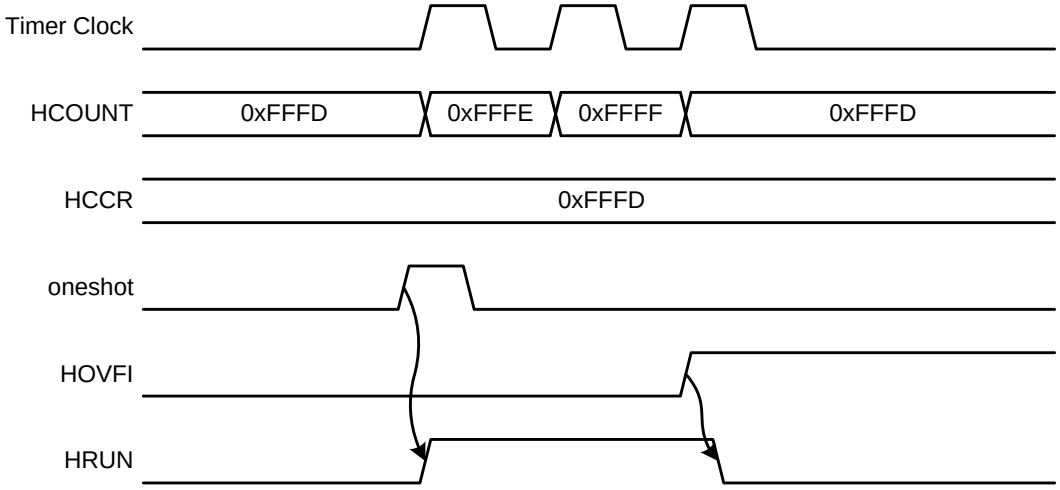


Figure 35.18. One Shot Mode Timing Diagram

35.5.8. Square Wave Output Mode

In square wave output mode, the timer operates the same as in auto-reload mode, except that the EX pin is toggled when an overflow event occurs. The state of the pin can be read or written using the HSTATE or LSTATE bit.

If operating in 16-bit mode, Equation 35.4 describes the high timer's output frequency in square wave output mode (the low timer cannot be used in 16-bit square wave output mode).

$$F_{OUT} = \frac{1}{2} \times \frac{F_{TIMER}}{65536 - HCCR}$$

Equation 35.4. 16-bit Square Wave Output Frequency

Equation 35.5 describes the timer's output frequency in square wave output mode if operating as a 32-bit timer.

$$F_{OUT} = \frac{1}{2} \times \frac{F_{TIMER}}{4294967296 - CAPTURE}$$

Equation 35.5. 32-bit Square Wave Output Frequency

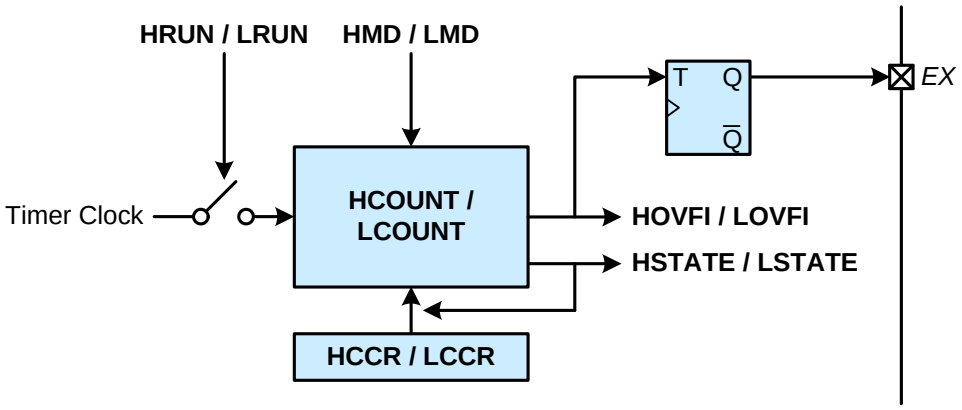


Figure 35.19. Square Wave Output Mode Block Diagram

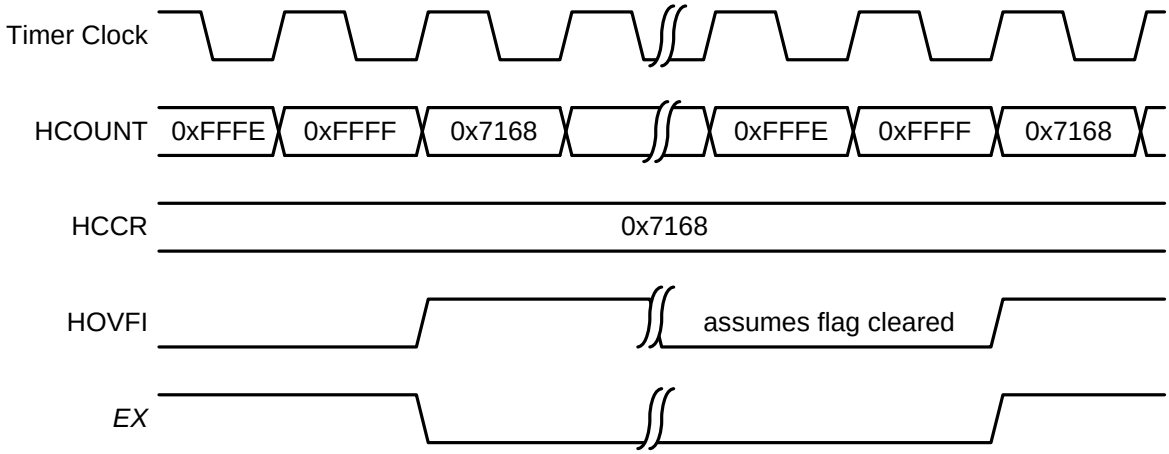


Figure 35.20. Square Wave Output Mode Timing Diagram

35.5.9. Pulse Width Modulation (PWM) Mode

In PWM mode, a PWM waveform is generated at the EX pin. The timer counts up to all 1's. Upon an overflow, the timer is loaded with all 0's, the pin is cleared to a low state, and the overflow flag is set. When the value of the count register is equal to the capture/compare/reload register, the EX pin is set to a high state and the HEXI or LEXI flag is set. It is possible to control the duty cycle of the waveform by writing to the HCCR or LCCR fields, and the EX pin can be read or written using the HSTATE or LSTATE bit.

If operating in 16-bit mode, Equation 35.6 describes the high timer's PWM duty cycle (the low timer cannot be used in 16-bit PWM mode).

$$\text{Duty Cycle} = \frac{65536 - \text{HCCR}}{65536}$$

Equation 35.6. 16-bit PWM Duty Cycle

Equation 35.7 describes the timer's output frequency in square wave output mode if operating as a 32-bit timer.

$$\text{Duty Cycle} = \frac{4294967296 - \text{CAPTURE}}{4294967296}$$

Equation 35.7. 32-bit PWM Duty Cycle

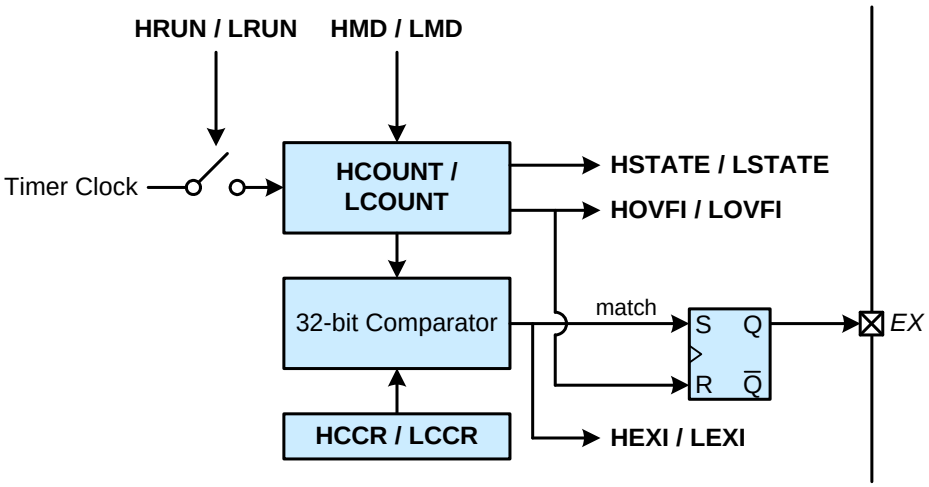


Figure 35.21. PWM Mode Block Diagram

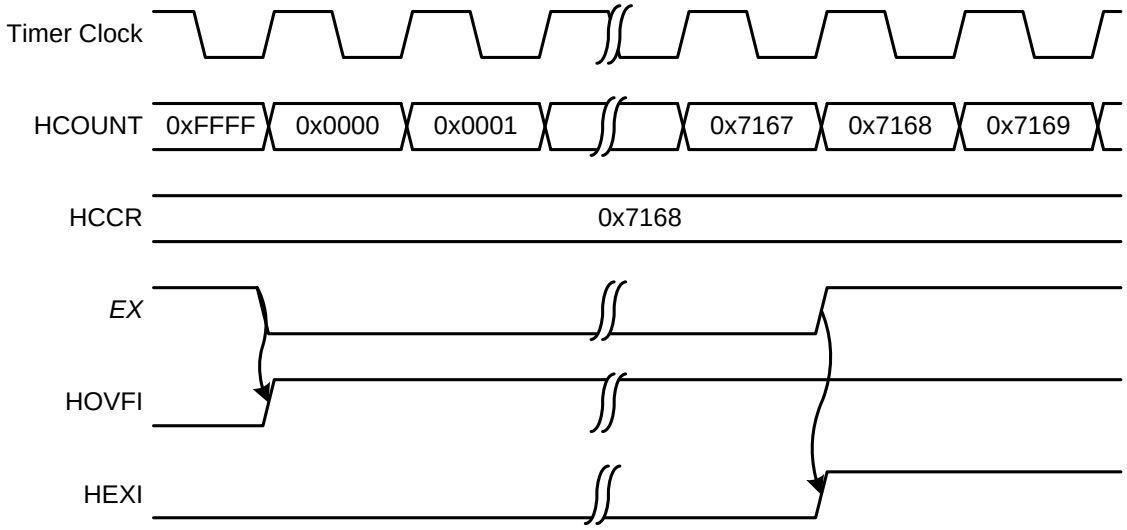


Figure 35.22. PWM Mode Timing Diagram

35.6. TIMER0, TIMER1 and TIMER2 Registers

This section contains the detailed register descriptions for TIMER0, TIMER1 and TIMER2 registers.

Register 35.1. TIMERn_CONFIG: High and Low Timer Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HOVFI	HEXI	HRUN	HSTATE	HMD				HOVFIEN	HEXIEN	DBGMD	HMSTREN	MSTRUN	Reserved	HCLK	
Type	RW	RW	RW	RW	RW				RW	RW	RW	RW	RW	R	RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LOVFI	LEXI	LRUN	LSTATE	Reserved	LMD			LOVFIEN	LEXIEN	SPLITEN	LMSTREN	Reserved		LCLK	
Type	RW	RW	RW	RW	R	RW			RW	RW	RW	RW	R		RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

TIMER0_CONFIG = 0x4001_4000
TIMER1_CONFIG = 0x4001_5000
TIMER2_CONFIG = 0x4001_6000
This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 35.2. TIMERn_CONFIG Register Bit Descriptions

Bit	Name	Function
31	HOVFI	**High Timer Overflow Interrupt Flag.** If split mode is enabled (SPLITEN = 1), this bit indicates the high 16-bit timer has wrapped or reloaded after reaching all 1's. If split mode is disabled (SPLITEN = 0), this value indicates the 32-bit timer has wrapped or reloaded after reaching all 1's. The timer module can set this bit in all modes. This bit must be cleared by software.
30	HEXI	**High Timer Extra Interrupt Flag.** This bit indicates the high 16-bit timer (or 32-bit timer if SPLITEN = 0) has been captured, reloaded with all 1's when counting down, or the timer matched the capture register in PWM mode. This interrupt flag can be set by the timer module in all modes except Auto-Reload and Toggle. This bit must be cleared by firmware.

Notes:
1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 35.2. TIMERn_CONFIG Register Bit Descriptions

Bit	Name	Function
29	HRUN	**High Run Control.** This bit is the run control for the high timer. When split mode is disabled (SPLITEN = 0), this bit also controls the low timer. 0: Stop the high timer or entire 32-bit timer. 1: The high timer runs if HMSTREN = 0 or MSTRUN (in Timer 0) = 1. The full 32-bit timer runs if split mode is disabled and (HMSTREN = 0 or MSTRUN = 1).
28	HSTATE	**High Multi Purpose State Indicator.** In PWM mode, the HSTATE bit is cleared each time the timer overflows from all 1's to all 0's. The HSTATE bit is then set when the timer increments while matching the capture register value. In Low/High/DC capture modes, the HSTATE bit indicates that the timer has captured due to the leading edge of TnEX. In all other modes, HSTATE is complemented each time the timer wraps or reloads when equal to all 1's (i.e. each time the HOVFI flag is also set). When in PWM or square wave output mode, the HSTATE bit defines the output state of the TnEX pin.
27:24	HMD	**High Timer Mode.** This field controls the mode of the high timer when split mode is enabled and the entire 32-bit timer when split mode is disabled. 0000: The high 16-bit timer or entire 32-bit timer is in Auto-Reload Mode. 0001: The high 16-bit timer or entire 32-bit timer is in Up/Down Count Mode. 0010: The high 16-bit timer or entire 32-bit timer is in Falling Edge Capture Mode. 0011: The high 16-bit timer or entire 32-bit timer is in Rising Edge Capture Mode. 0100: The high 16-bit timer or entire 32-bit timer is in Low Time Capture Mode. 0101: The high 16-bit timer or entire 32-bit timer is in High Time Capture Mode. 0110: The high 16-bit timer or entire 32-bit timer is in Duty Cycle Capture Mode. 0111: The high 16-bit timer or entire 32-bit timer is in Oneshot Mode. 1000: The high 16-bit timer or entire 32-bit timer is in Square Wave Output Mode. 1001: The high 16-bit timer or entire 32-bit timer is in PWM Mode. 1010-1111: Reserved.
23	HOVFIEN	**High Timer Overflow Interrupt Enable.** 0: The state of HOVFI does not affect the high timer interrupt. 1: A high timer interrupt request is generated if HOVFI is set to 1.
22	HEXIEN	**High Timer Extra Interrupt Enable.** 0: The state of the HEXI flag does not affect the high timer interrupt. 1: A high timer interrupt request is generated if HEXI is set to 1.
21	DBGMD	**Timer Debug Mode.** 0: The timer will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the Timer to halt.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 35.2. TIMERn_CONFIG Register Bit Descriptions

Bit	Name	Function
20	HMSTREN	**High Master Enable.** This bit determines whether the run master control (MSTRUN in the master TIMER0 module) must be set in addition to the timer's run bit (HRUN) before the high timer will run. This controls the entire 32-bit timer in 32-bit mode. 0: MSTRUN does not need to be set for the timer to run. 1: MSTRUN must be set for the timer to run.
19	MSTRUN	**Master Run Control.** MSTRUN is connected from the master timer to all slave timers. This bit is only used in the master timer (TIMER0). 0: Disable the master run control for all timers. 1: Enable the master run control for all timers.
18	Reserved	Must write reset value.
17:16	HCLK	**High Clock Source.** Selects the clock source of the high 16-bit timer if split mode is enabled (SPLITEN = 1) and the entire 32-bit timer if split mode is disabled (SPLITEN = 0). 00: Select the APB clock as the timer source. 01: Select the external oscillator clock as the timer source. The external oscillator must run slower than one-half the APB clock. 10: Select the dedicated 8-bit prescaler as the timer source. 11: Select falling edges of the CT signal as the timer clock source.
15	LOVFI	**Low Timer Overflow Interrupt.** The hardware sets this bit when the low 16-bit timer has wrapped or reloaded after reaching all 1's. This bit is set by the module regardless of the state of SPLITEN and can be set in all modes. This bit must be cleared by firmware.
14	LEXI	**Low Timer Extra Interrupt Flag.** This bit is set by hardware when the low 16-bit timer has been captured, reloaded with all 1's when counting down, or the timer matched the capture register in PWM mode. This interrupt flag can be set by hardware in all modes except Auto-Reload and Square Wave. This flag is not set by hardware when split mode is disabled (SPLITEN = 0). This bit must be cleared by firmware.
13	LRUN	**Run Control Low.** LRUN is the run control for the Low Timer in split mode. 0: Stop the low timer if split mode is enabled (SPLITEN = 1). 1: The low timer runs if split mode is enabled (SPLITEN = 1) and (LMSTREN = 0 or MSTRUN = 1 in Timer 0).
12	LSTATE	**Low Multi Purpose State Indicator.** In Low/High/DC capture modes, the LSTATE bit indicates that the low timer has captured due to the leading edge of TnEX. In all other modes, LSTATE is complemented each time the timer wraps or reloads when equal to all 1's (i.e. each time the LOVFI flag is also set).

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 35.2. TIMERn_CONFIG Register Bit Descriptions

Bit	Name	Function
11	Reserved	Must write reset value.
10:8	LMD	**Low Timer Mode.** This field controls the mode of the low timer when split mode is enabled (SPLITEN = 1). 000: The low timer is in Auto-Reload Mode. 001: The low timer is in Up/Down Count Mode. 010: The low timer is in Falling Edge Capture Mode. 011: The low timer is in Rising Edge Capture Mode. 100: The low timer is in Low Time Capture Mode. 101: The low timer is in High Time Capture Mode. 110: The low timer is in Duty Cycle Capture Mode. 111: The low timer is in Oneshot Mode.
7	LOVFIEN	**Low Timer Overflow Interrupt Enable.** 0: The state of LOVFI does not affect the low timer interrupt. 1: A low timer interrupt request is generated if LOVFI = 1.
6	LEXIEN	**Low Timer Extra Interrupt Enable.** 0: The state of the LEXI flag does not affect the low timer interrupt. 1: A low timer interrupt request is generated if LEXI is set to 1.
5	SPLITEN	**Split Mode Enable.** This bit selects between a single 32-bit timer or two 16-bit timers. 0: The timer operates as a single 32-bit timer controlled by the high timer fields. 1: The timer operates as two independent 16-bit timers.
4	LMSTREN	**Low Run Master Enable.** This bit determines whether the run master control (MSTRUN in the master TIMER0 module) must be set in addition to the timer's run bit (LRUN) before the low timer will run. 0: MSTRUN does not need to be set for the low timer to run. 1: MSTRUN must be set for the low timer to run.
3:2	Reserved	Must write reset value.
1:0	LCLK	**Low Clock Source.** Select the clock source for the low 16-bit timer if SPLITEN is set to 1. If SPLITEN is cleared to 0, the low timer source is controlled by HCLK. 00: Select the APB clock as the timer source. 01: Select the external oscillator clock as the timer source. The external oscillator must run slower than one-half the APB clock. 10: Select the dedicated 8-bit prescaler as the timer source. 11: Select falling edges of the CT signal as the timer clock source.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 35.2. TIMERn_CLKDIV: Module Clock Divider Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								CLKDIVCT							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								CLKDIVRL							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

TIMER0_CLKDIV = 0x4001_4010
TIMER1_CLKDIV = 0x4001_5010
TIMER2_CLKDIV = 0x4001_6010

Table 35.3. TIMERn_CLKDIV Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:16	CLKDIVCT	**Clock Divider Counter.** This field provides direct access to the 8-bit prescaler counter. The counter repeatedly counts from the value in CLKDIVRL to all 1's.
15:8	Reserved	Must write reset value.
7:0	CLKDIVRL	**Clock Divider Reload Value.** This field holds the reload value for the 8-bit clock divider. The clock divider output frequency is given by: $F_{TIMER} = \frac{F_{APB}}{(256 - CLKDIVRL)}$

Register 35.3. TIMERn_COUNT: Timer Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HCOUNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LCOUNT															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

TIMER0_COUNT = 0x4001_4020
TIMER1_COUNT = 0x4001_5020
TIMER2_COUNT = 0x4001_6020

Table 35.4. TIMERn_COUNT Register Bit Descriptions

Bit	Name	Function
31:16	HCOUNT	**High Timer Count.** This field holds the high timer state.
15:0	LCOUNT	**Low Timer Count.** This field holds the low timer state.

Register 35.4. TIMERn_CAPTURE: Timer Capture/Reload Value

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	HCCR															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	LCCR															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Addresses

TIMER0_CAPTURE = 0x4001_4030
TIMER1_CAPTURE = 0x4001_5030
TIMER2_CAPTURE = 0x4001_6030

Table 35.5. TIMERn_CAPTURE Register Bit Descriptions

Bit	Name	Function
31:16	HCCR	**High Timer Capture/Reload.** This field holds the high timer capture or reload value.
15:0	LCCR	**Low Timer Capture/Reload.** This field holds the low timer capture or reload value.

35.7. TIMERn Register Memory Map

Table 35.6. TIMERn Memory Map

Register Name	TIMERn_CAPTURE	TIMERn_COUNT	TIMERn_CLKDIV	TIMERn_CONFIG																											
ALL Offset	0x30	0x20	0x10	0x0																											
Access Methods	ALL	ALL	ALL	ALL | SET | CLR	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	HCCR	HCOUNT	Reserved	HOVFI	HEXI	HRUN	HSTATE	HMD				HOVFIEN	HEXIEN	DBGMD	HMSTREN	MSTRUN	Reserved	HCLK															
			CLKDIVCT																														
	LCCR	LCOUNT	Reserved	LOVFI	LEXI	LRUN	LSTATE	Reserved	LMD																								
			CLKDIVRL																									LOVFIEN	LEXIEN	SPLITEN	LMSTREN	Reserved	LCLK		

Notes:

1. The "ALL Offset" refers to the address offset of the ALL access method for a register, this offset should be referenced to the base address for the block. For example, if a register block has a base address of 0x4001_0000 and the ALL offset is specified to be 0xA4, the register's absolute ALL access address is located at 0x4001_00A0 in the address map. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. The register with ALL access at 0x4001_00A0 may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.
2. The base addresses for this register block are: TIMER0 = 0x4001_4000, TIMER1 = 0x4001_5000, TIMER2 = 0x4001_6000

36. Universal Synchronous/Asynchronous Receiver/Transmitter (USART0)

This section describes the USART module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the USART block, which is used by USART0 on all device families covered in this document.

36.1. USART Features

The USART module includes the following features:

- Independent transmitter and receiver configurations with separate 16-bit baud rate generators.
- Synchronous or asynchronous transmissions and receptions.
- Clock master or slave operation with programmable polarity and edge controls.
- Up to 5 Mbaud (synchronous or asynchronous, TX or RX, and master or slave) or 1 Mbaud Smartcard (TX or RX).
- Individual enables for generated clocks during start, stop, and idle states.
- Internal transmit and receive FIFOs with flush capability and support for byte, half-word, and word reads and writes.
- Data bit lengths from 5 to 9 bits.
- Programmable inter-packet transmit delays.
- Auto-baud detection with support for the LIN SYNC byte.
- Automatic parity generation (with enable).
- Automatic start and stop generation (with separate enables).
- Transmit and receive hardware flow-control.
- Independent inversion correction for TX, RX, RTS, and CTS signals.
- IrDA modulation and demodulation with programmable pulse widths.
- Smartcard ACK/NACK support.
- Parity error, frame error, overrun, and underrun detection.
- Multi-master and half-duplex support.
- Multiple loop-back modes supported.
- Multi-processor communications support.

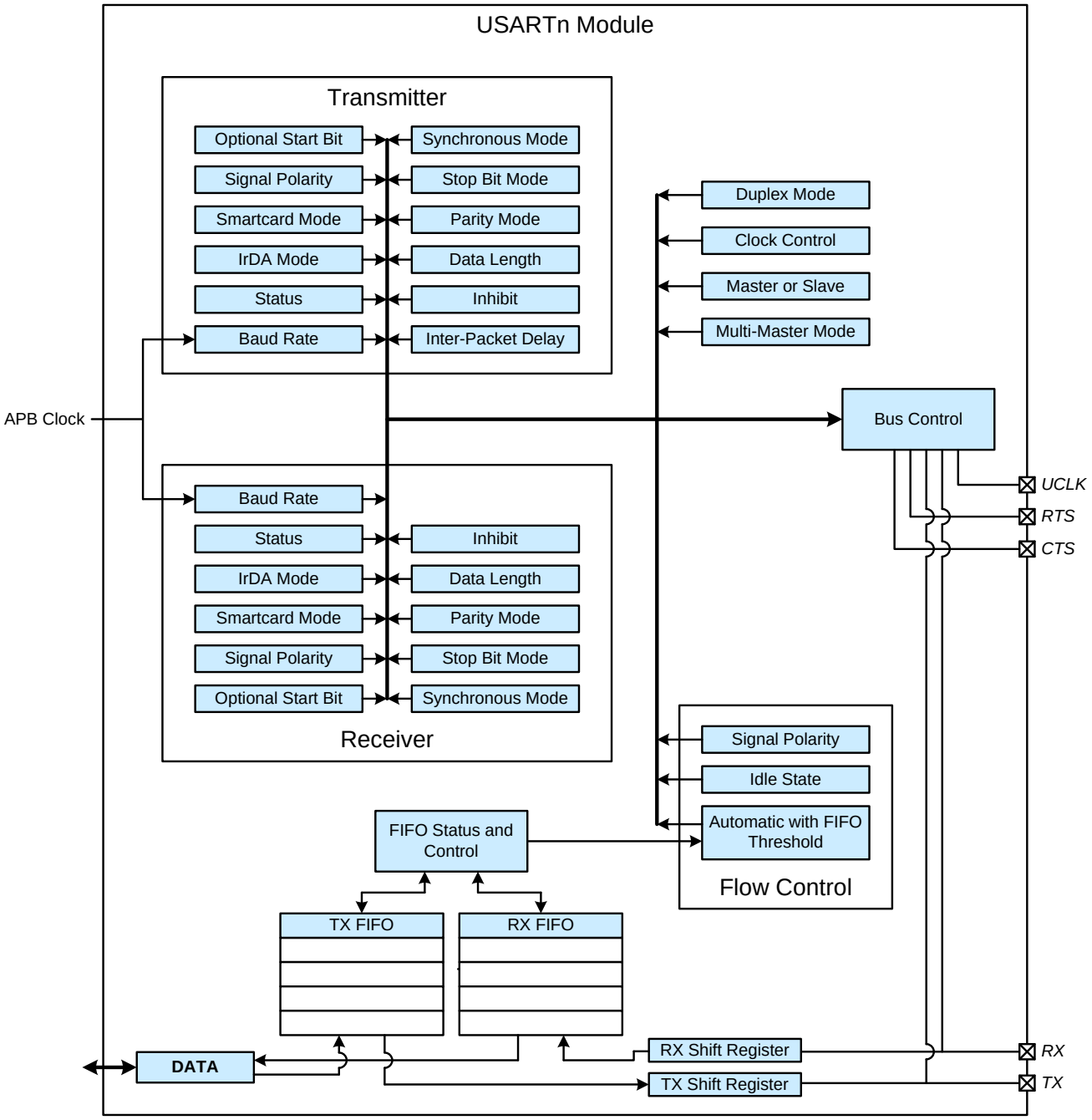


Figure 36.1. USART Block Diagram

36.2. Basic Data Format

The USART module data consists of four parts: start bit, data, parity bit, and stop bit.

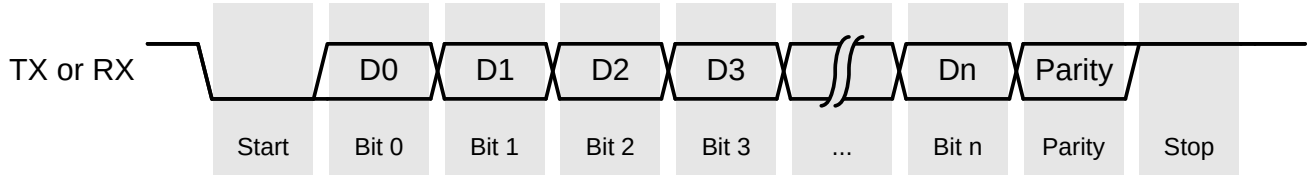


Figure 36.2. Basic Asynchronous USART Communication

Start bits are optional and can be enabled using the transmitter start enable (TSTRTEN) and receiver start enable (RSTRTEN) bits. Disabling the start bits may be useful in certain synchronous protocols, but is not generally used for asynchronous operation.

The data length is variable and can be set to 5 to 9 bits using the transmitter data length (TDATLN) or receiver data length (RDATLN) bit fields.

The transmitter will transmit a parity bit when the transmitter parity enable (TPAREN) bit is set. The transmitter parity mode (TPARMD) field configures the parity for the transmit data as either odd, even, set (parity always set to 1), or clear (parity always cleared to 0).

Similarly, the receiver will expect to receive a parity bit when receiver parity is enabled (RPAREN = 1). The RPARMD field controls the receiver parity mode and sets it to odd, even, set (parity always expected to be set to 1), or clear (parity always expected to be cleared to 0).

The stop bits are also fully configurable using the transmitter stop enable (TSTPEN) and receiver stop enable (RSTPEN) bits. The transmitter stop mode (TSTPMD) and receiver stop mode (RSTPMD) fields configure the stop length to 0.5, 1, 1.5, or 2 bits. Transmitting partial stop bits (0.5 or 1.5) causes subsequent transmitted bits to slip by one-half of a bit-time.

36.3. Baud Rate

The USART can operate as two independent, one-way channels since the transmitter and the receiver each have their own internal baud rate generators, FIFOs and shift registers. For example, the transmitter can operate in synchronous mode at one baud rate while the receiver operates in asynchronous mode at a different baud rate.

If only one-half of the USART module operates in synchronous mode, the associated baud rate generator controls the clock frequency. If both the transmitter and receiver operate in synchronous mode, then the transmitter's baud rate generator controls the UCLK frequency.

In asynchronous or synchronous master modes, the transmitter baud rate control (TBAUD) bit field sets the transmitter baud rate. In asynchronous master mode, the receiver baud rate control (RBAUD) bit field sets the receiver baud rate. The receiver baud rate control (RBAUD) bit field sets the receiver baud rate in synchronous master mode only if the transmitter is not also in synchronous master mode. The equations in the TBAUD and RBAUD field descriptions determine the transmitter or receiver baud rate. The associated baud rate generator is unused in synchronous slave mode.

36.3.1. Receiver Auto-Baud Detection

The hardware receiver auto-baud detection automatically derives the appropriate RBAUD value from the received data. When receive auto-baud detection is enabled (RABDEN = 1), the hardware requires the first incoming data to be a value that guarantees an RX state transition between each bit: 0x55 when in non-IrDA mode and 0x00 when in IrDA mode. The receiver auto-baud detection supports either 8- or 9-bit data lengths with any combination of parity or stop bits.

Figure 36.3 shows the USART receive auto-baud detection input timing.

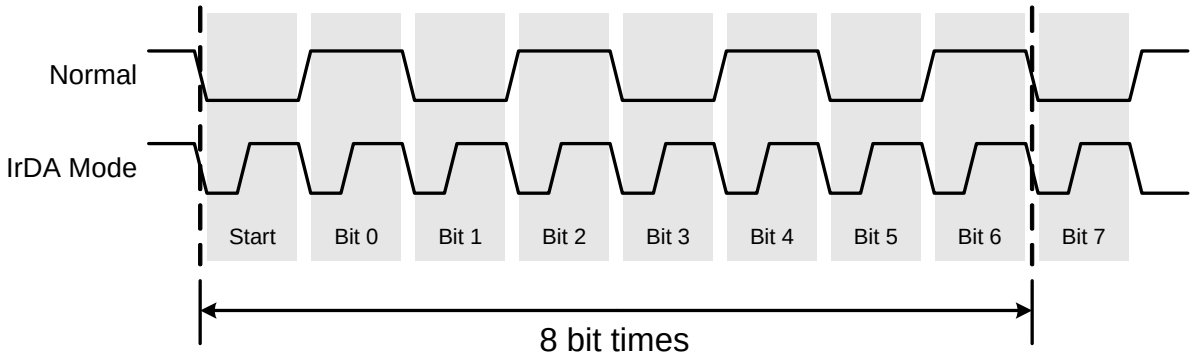


Figure 36.3. Receiver Auto-Baud Detection Input Timing

The receiver auto-baud detection calculates the appropriate value for RBAUD by counting the number of clock cycles between the leading edge of the start bit and the leading edge of the 8th data bit and dividing the clock cycle count by either 16 (when RIRDAEN = 0) or 128 (when RIRDAEN = 1). The hardware stores the calculated value to RBAUD and clears the receive auto-baud detection enable bit (RABDEN = 0). The new baud rate is used starting at the 8th bit of the incoming packet. The receiver will treat the received data (0x55 or 0x00) like any other received data and check it for parity errors and store it in the FIFO. If the first incoming byte causes the RBAUD value to overflow, indicating an auto-baud error, hardware will set the receive frame error interrupt (RFRMERI) flag regardless of the sampled state of the stop bits.

36.4. Interrupts

The USART module contains several interrupt sources that cause a vector to the USART interrupt. All of the interrupt flags may be masked by clearing a corresponding interrupt enable bit.

36.4.1. Transmit Interrupt Sources

The transmit data request interrupt (TDREQI) can be enabled by setting the TDREQIEN bit to 1. This flag indicates that the transmitter is requesting more FIFO data. The TDREQI flag automatically clears when the number of full entries in the transmit FIFO is more than the TFTH setting.

The transmit complete interrupt (TCPTI) flag indicates the following:

1. If TCPTTH = 0, this flag indicates a single transmit completed since TCPTI was last cleared.
2. If TCPTTH = 1, this flag indicates a transmit of the last available data in the FIFO completed since TCPTI was last cleared.

In either scenario, the hardware will generate an interrupt if the TCPTIEN bit is set to 1 when TCPTI is set.

The transmitter has two error condition flags. The underrun error interrupt (TUREI) flag indicates when the transmitter is required to send data but no data is available in the transmit FIFO. The Smartcard parity error interrupt (TSCERI) flag sets when the hardware detects the TX pin state as 0 at the end of the first whole stop bit period when operating in Smartcard mode (TSCEN = 1). Both of these interrupts are enabled with the transmit error interrupt enable (TERIEN) bit. Transmissions are automatically inhibited when any enabled error flag causes the interrupt to assert.

Finally, the transmitter has a transmit FIFO error interrupt (TFERI) flag which is set whenever an illegal FIFO write (e.g., a write to a full FIFO) is detected. Note that TFERI does not have an interrupt enable/disable control bit, so if the TFERI flag is set an interrupt will always be generated (assuming the USART module interrupt is enabled).

36.4.2. Receive Interrupt Sources

The receive data request interrupt (RDREQI) flag indicates that at least the number of receive FIFO slots set by the receive FIFO threshold (RFTH) are full, and firmware can read data from the receive FIFO. This interrupt is enabled by setting the receive data request interrupt enable (RDREQIEN) to 1. This flag automatically clears when the number of filled slots in the receive FIFO drops below the RFTH setting.

The receiver has three error interrupt sources. The receive overrun error interrupt (ROREI) flag indicates when the receive FIFO and shift register is full, and the shift register has been erroneously overwritten by additional receive data. The receive parity error interrupt (RPARERI) indicates that an invalid parity bit has been received. Finally, the receive frame error interrupt (RFRMERI) indicates when an expected whole stop bit is low, when a frame match mode fails, or when an auto-baud error occurs. The receive error interrupt enable (RERIEN) bit enables these flags as interrupt sources.

Finally, the receiver has a receive FIFO error interrupt (RFERI) flag which is set whenever an illegal FIFO read (e.g, a read from an empty FIFO) is detected. Note that RFERI does not have an interrupt enable/disable control bit, so if the RFERI flag is set an interrupt will always be generated (assuming the USART module interrupt is enabled).

36.5. Flow Control

36.5.1. Transmitter Hardware Flow Control

The CTS enable (CTSEN) bit enables hardware flow control in the transmitter. When CTS is enabled, the transmitter will only begin transmissions if the CTS input is low after the optional inversion using the CTSINVEN bit.

Firmware can read the current state of the CTS pin using the CTS bit. The CTS value read will be inverted from the CTS pin if the signal is inverted (CTSINVEN = 1).

36.5.2. Receiver Hardware Flow Control

Firmware can set the RTS enable (RTSEN) bit to enable hardware flow control in the receiver. When flow control is enabled, the receiver asserts RTS under any of the following conditions:

1. The receiver is disabled (REN = 0).
2. The receiver is inhibited (RINH = 1).
3. The receiver is stalled during a debug halt (DBGMD = 1).
4. An error flag sets causing the receive error interrupt to assert (RERIEN = 1 and RFRMERI, ROREI, or RPARERI is also set).
5. The receiver can store zero (if RTSTH = 0) or up to one (if RTSTH = 1) more data.

The RTS signal holds its last value when disabled by clearing RTSEN to 0. The RTS bit allows firmware to read the state of the RTS signal. Firmware can also write the RTS bit when flow control is disabled (RTSEN = 0) to set the state of the RTS pin. Note that the value written into the RTS bit will be inverted at the RTS pin if the RTS invert enable (RTSINVEN) bit is set to 1.

36.5.3. Inter-Packet Delay Generator

The transmitter supports a configurable delay between transmissions that may be used to limit the transmission rate in applications that do not support hardware flow control.

The hardware calculates the inter-packet delay using a counter operating at the current baud rate. As a result, the IPDELAY field represents multiples of the transmitter bit times. For example, setting the inter-packet delay (IPDELAY) field to 5 results in a delay equal to 5 bit times. Setting the IPDELAY field to 0 disables the inter-packet delay.

If in synchronous slave mode, UCLK must be running to use the inter-packet delay feature.

36.6. Debug Mode

Firmware can set the DBGMD bit to force the USART module to halt on a debug breakpoint. The module will complete any active transmissions and receptions before stopping. Clearing the DBGMD bit forces the USART module to continue operating while the core halts in debug mode. Note that when IPDELAY is set to zero, the USART may not halt until the FIFO is empty. If IPDELAY is non-zero, the USART will complete the current transfer but will not empty the FIFO.

36.7. Sending Data

To begin a transmit operation, firmware should set all the necessary transmitter configuration bits, enable the transmitter (TEN = 1), and write the outgoing data to the DATA register. Hardware will automatically transfer the data from the DATA register to the transmit FIFO. A FIFO entry is immediately loaded from the FIFO into the shift register any time data is available in the transmit FIFO and the shift register is empty.

The data transmission begins when all of the following conditions are met:

1. The shift register is loaded with data.
2. The transmitter is enabled (TEN = 1).
3. The transmitter is not inhibited (TINH = 0).
4. The CTS pin is deasserted if flow control is enabled (CTSEN = 1).
5. The module is not halted because of a debug breakpoint if DBGMD is set to 1.
6. The TX output enable is set (TXOEN = 1).

36.7.1. Transmitter FIFO Management

The USART transmit FIFO is implemented as a circular buffer with four entries, allowing data to be pushed as a continuous stream. The DATA register forms a single port into the transmit and receive FIFOs. Writes to the DATA register push data into the transmit FIFO, and reads from the DATA register pop data from the receive FIFO.

The transmitter supports byte, half-word, or word writes to the FIFO. Writes to the FIFO should always be right-justified, so byte-wide writes should always write DATA[7:0], half-word writes should always write DATA[15:0], and word writes should always write DATA[31:0]. The transmitter ignores all other writes to the DATA register, including left-justified byte writes or writes containing more data than will fit in the empty slots in the FIFO. Any illegal FIFO write sets the transmit FIFO error interrupt flag (TFERI) and causes an interrupt, if enabled.

The TDATLN bit controls how the written data is mapped into the FIFO. The transmitter supports data lengths ranging from 5 to 9 bits; the optional start, parity, and stop bits are not stored in the FIFO. When transmitting less than 9 bits of data, the hardware transmits all 5 to 8 bits written to the FIFO by firmware.

When transmitting 9-bit data, the transmitter supports two modes of operation:

1. In normal 9-bit mode (TDATLN = 4), firmware writes 9-bit data to the FIFO with each half-word containing a right-justified 9-bit entry.
2. In fixed 9-bit mode (TDATLN = 5), bus writes to the FIFO are assumed to contain only the least-significant 8 bits of each data entry, and hardware inserts the 9th bit value (set by TBIT) into the FIFO on each FIFO write.

When the data length is 8 bits (or 9 bits when using TDATLN = 5), one, two, or four FIFO entries can be written in a single bus operation by using byte, half-word, or word operations, respectively. For example, writing a whole word (4 bytes) to DATA[31:0] will write four entries in the FIFO with each byte as an entry.

When the data length is 9 bits (using TDATLN = 4), two FIFO entries can be written in a single bus operation by using a word operation, or one entry can be written using a half-word operation. With a half-word write, the hardware will push DATA[8:0] to the FIFO. With a word write, the hardware will push DATA[8:0] to the FIFO, followed by the value in DATA[24:16]. The other bits in the DATA register in this mode are unused and have no effect.

The transfer FIFO count (TCNT) field maintains a count of the number of occupied entries in the transmit FIFO. Firmware may read TCNT to determine the allowable width of writes (byte, half-word, or word) given the available number of empty FIFO entries.

The transmit FIFO threshold (TFTH) field configures the threshold at which the hardware asserts the transmit data request interrupt (TDREQI) or DMA data request and may be configured to 1, 2, or 4 empty FIFO slots.

Firmware can set the transmit FIFO flush (TFIFOFL) bit to flush the contents of the transmit FIFO. A FIFO flush will also clear the contents of the shift register if the transmitter is idle and is not actively transmitting data.

36.7.2. Transmitter Status

The USART module has two status bits which may be monitored to determine the status of the transmitter.

36.7.2.1. Transmitter Busy Flag (TBUSYF)

The transmitter busy flag (TBUSYF) will be asserted when a single byte transmission is in progress. Note that if transmitted multiple bytes with IPDELAY>0, TBUSYF will return to 0 between each transmitted byte. For this reason, TBUSYF is not a reliable indicator of a completed transmission, and the method described below using the TCPTI bit is recommended.

36.7.2.2. Transmitter Complete Interrupt Flag (TCPTI)

The transmit complete interrupt (TCPTI) flag indicates the following:

1. If TCPTTH = 0, this flag indicates a single transmit completed since TCPTI was last cleared.
2. If TCPTTH = 1, this flag indicates a transmit of the last available data in the FIFO completed since TCPTI was last cleared.

To determine when the transmitter has completed transmission of all data in the FIFO, firmware may set TCPTTH=1, write the data into the FIFO, and then wait for the TCPTI bit to be asserted. Note that TCPTI must be manually cleared by firmware.

36.8. Receiving Data

Firmware should first configure all necessary receiver configuration bits before enabling the receiver (REN = 1). Incoming data will be de-glitched and then shifted into the receive shift register. When the data reception is complete, the data will be aligned and padded according to the data length specified in the RDATLN bit field, and then pushed into the circular receive FIFO buffer if an empty entry is available. When both the receive FIFO and the shift register are full, any subsequent received data replaces the shift register contents and causes the hardware to set the receive overrun error interrupt (ROREI) flag.

Firmware may read from the FIFO any time entries are available regardless of the state of the receiver.

The receiver supports one-shot receptions, where the receiver accepts a single byte. Firmware can enable this by setting the ROSEN bit. The hardware automatically clears this bit after the next reception is completed and accepted. If MATMD is set to 1 for MCE mode and a match does not occur, the hardware will not accept the data and will not clear the ROSEN bit.

36.8.1. Receiver FIFO Management

The receive FIFO is implemented as a circular buffer with four entries. The receive and transmit FIFOs are both accessed using the same DATA register; a read from the DATA register pops data from the receive FIFO, and a write to the DATA register pushes data into the transmit FIFO. The receiver supports byte, half-word, or word DMA and bus reads from the FIFO. When reading from the FIFO, the least-significant entry is popped from the FIFO first.

The RDATLN field determines how the data is mapped into the FIFO. The receiver supports data lengths ranging from 5 to 9 bits; the optional start, parity, and stop bits are not stored in the FIFO. When receiving less than 9 bits, a firmware read from the FIFO returns all 5 to 8 bits of the receive data.

When receiving 9-bit data, the receiver supports two modes of operation:

1. In normal 9-bit mode (RDATLN = 4), firmware reads 9-bit data from the FIFO, with each half-word containing a right-justified 9-bit entry.
2. In fixed 9-bit mode (RDATLN = 5), only the least-significant 8 bits of received data are stored in the FIFO. The 9th bit is compared against RBIT to generate an interrupt or set error flags according to the MATMD field.

When the data length is 8 bits (or 9 bits when RDATLN = 5), firmware can read one, two, or four FIFO entries in a single bus operation by using byte, half-word, or word operations, respectively.

When the data length is 9 bits (using normal 9-bit mode with RDATLN = 4), each half-word is treated as a right-justified entry. When reading 9-bit data using a half-word bus operation, the hardware places the first entry popped from the FIFO into DATA[8:0]. When reading a whole word, hardware places the first entry popped from the FIFO into DATA[8:0] and the second entry into DATA[24:16]. The rest of the DATA bits are unused in this mode and are set to zero.

The receive FIFO count (RCNT) field maintains a count of the number of occupied entries in the receive FIFO. Firmware may read RCNT to determine the allowable read widths given the available number of full FIFO entries. The receive FIFO threshold (RFTH) field configures the threshold at which hardware asserts a read request interrupt (RDREQI) or DMA data request and may be configured to 1, 2, or 4 occupied entries.

Firmware can set the receive FIFO flush (RFIFOFL) bit to flush the contents of the receive FIFO. A FIFO flush will also clear the contents of the shift register if the receiver is idle and is not actively receiving data.

36.8.2. Receiver Status

The USART module includes a status bit which may be monitored to determine the status of the receiver.

36.8.2.1. Receiver Busy Flag (RBUSYF)

The receiver busy flag (RBUSYF) will be asserted when a single byte receive is in progress. Note that when receiving multiple bytes, any delay between the bytes may cause RBUSYF to return to 0 between each received byte.

36.9. Synchronous Communications

Synchronous communication is similar to basic asynchronous protocol with the addition of a synchronous UCLK clock signal. The UCLK signal is an output in master mode and an input when operating as a slave.

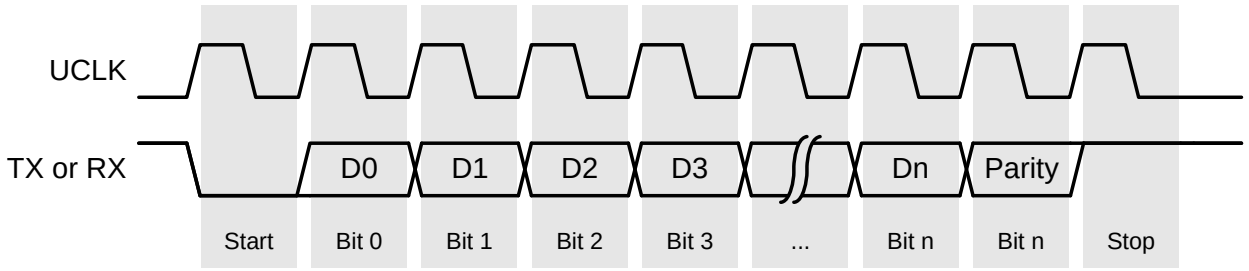


Figure 36.4. Synchronous USART Communication

Firmware can configure the USART transmitter or receiver for synchronous mode by setting the transmitter synchronous mode (TSYNCEN) or receiver synchronous mode (RSYNCEN) enable bits.

The transmitter and receiver share a common clock generator and will share a common UCLK pin and clock configuration if both are configured for synchronous operation. Either the transmitter or receiver may request that the hardware generate a clock when configured as a synchronous clock master. If only one-half of the module is operating in synchronous mode, that half's associated baud rate generator controls the clock frequency. If both the transmitter and receiver are operating in synchronous mode, then the transmitter's baud rate generator controls the UCLK frequency.

When both the transmitter and receiver are disabled or in asynchronous mode, the UCLK pin state may be read or written via the UCLK bit in the FLOWCN register.

36.9.1. Synchronous Clock Master Mode

The operational mode (OPMD) configures the USART module as either the synchronous master or slave. Setting this bit to 1 enables the module as the synchronous master.

By default, the master will hold the UCLK pin idle between transmissions. Firmware can set the idle clock control bit (ISTCLK) to 1 to generate a clock between transmissions.

To configure the transmitter to generate a clock during the start or stop bits, set the start state clock control (STRTSTCLK) or the stop state clock control (STPSTCLK) bits to 1.

36.9.2. Synchronous Clock Slave Mode

In synchronous clock slave mode, the UCLK pin is configured as an input, and the receiver or transmitter expect to receive a clock signal from the synchronous master.

If the transmitter is configured as a synchronous slave (OPMD = 0), is not configured to transmit a start bit (TSTRTEN = 0), and receives a UCLK edge causing it shift out a data bit with no data available, the transmitter will set the underrun error flag (TUREI) and generate an interrupt, if enabled (TERIEN = 1).

36.9.3. Synchronous Clock Configuration

The synchronous clock generator supports configurable UCLK polarity and phase using two control bits: sampling clock edge select (CLKESEL) and clock idle state (CLKIDLE). In synchronous master mode (OPMD = 1), these bits determine the characteristics of the generated clock; in synchronous slave mode (OPMD = 0), these bits determine the expected characteristics of the received clock.

The CLKESEL bit determines whether the generated or received clock falls (CLKESEL = 0) or rises (CLKESEL = 1) in the middle of each transmitted bit. The CLKIDLE bit determines whether the idle state of the generated or received clock is low (CLKIDLE = 0) or high (CLKIDLE = 1).

Figure 36.5 illustrates these clock configurations.

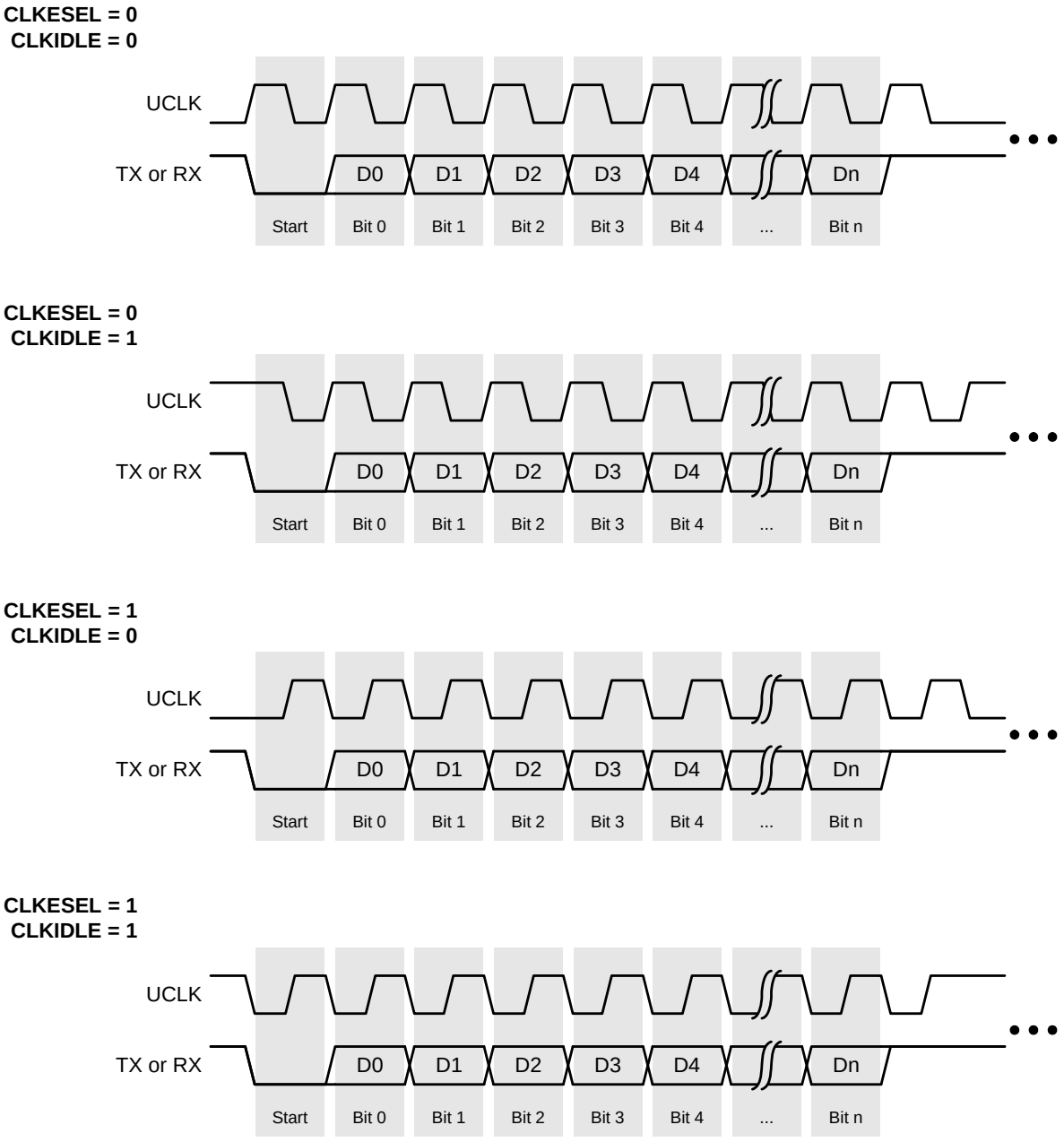


Figure 36.5. USART Clock Configurations

36.10. Additional Communication Support

36.10.1. Multi-Master Mode

The transmitter supports applications in which multiple transmitters are driving the same TX and UCLK signals. This multi-master mode is suited for applications using a Smartcard or LIN, which perform one-wire, half-duplex communications.

Multi-master mode requires the use of idle signal tristates (ITSEN = 1). When this bit is set to 1 and the transmitter is idle, the transmitter automatically tristates the TX pin when not transmitting. The transmitter will also automatically tristate the UCLK pin if the transmitter is the clock master. The TX (and UCLK pin, if the transmitter is the synchronous master) pin must have an external pull-up resistor to ensure the pin remains in an idle state when tristated.

Because a synchronous master will always attempt to drive a clock when it sees data on its RX input, if using multi-master mode in a synchronous application, the transmitting USART must set itself to slave mode between transmissions to prevent contention on the clock line.

36.10.2. Multi-Processor Communications

In a multi-processor application (MCE mode), a master first transmits an address byte to select the target. In an address byte, the 9th bit is always set to a logic 1; in a data byte, the 9th bit is always set to logic 0. The USART module supports multi-processor communication through the use of match operations. Although the match operations are most commonly used with the 9th data bit as an address indicator, the match operations are available regardless of the actual configured data length.

Match operations are enabled or disabled in the receiver by configuring the match mode (MATMD) bit field.

When MATMD is set to 1, the hardware operates in MCE mode and only accepts and stores the incoming receive data if the last data bit matches the value of RBIT. Otherwise, the hardware ignores incoming data, no interrupts are generated, and no FIFO activity occurs. This mode can be used to only accept data when the last data bit marks the frame as containing an address value. Firmware would then compare this address against an assigned value and reconfigure the receiver as needed if the values match. The MATMD setting is automatically switched to Frame mode after a match occurs in the MCE setting.

In Frame mode (MATMD = 2), all incoming data is accepted and stored, but a receive frame error (RFRMERI) asserts if the last data bit matches the value of RBIT. This operation can be used to assert an error when an unexpected address frame arrives.

In Store mode (MATMD = 3), the hardware accepts all incoming data and stores the last data bit in the RBIT field. The last data bit may also be stored in the FIFO depending on the value of RDATLN. For example, if RDATLN is set to 9 bits with the 9th bit stored in the FIFO (RDATLN = 4), and MATMD is set to store, the 9th data bit will be stored in both RBIT and the FIFO.

36.10.3. IrDA

The USART module transmitter and receiver support simple, asynchronous communications with IrDA devices.

36.10.3.1. IrDA Modulation

Firmware can place the transmitter in IrDA mode by setting the TIRDAEN bit. In IrDA mode, the transmitter performs a RZ (return-to-zero) modulation on the TX signal.

By default, transmitting a 0 causes a pulse to be generated low for a fraction of the bit time and transmitting a 1 leaves TX high without generating a low pulse. The transmit IrDA pulse width (TIRDAPW) bit field configures the pulse width as either 1/16, 1/8, 3/16, or 1/4 of the bit time.

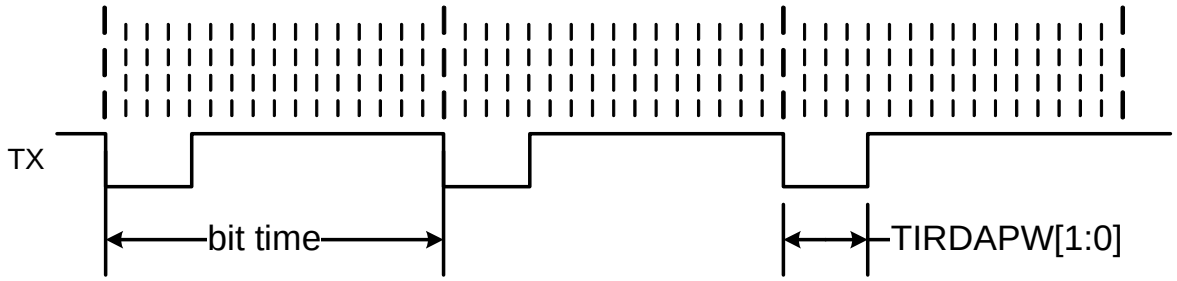


Figure 36.6. USART Transmit IrDA Pulse Width Timing

As shown in Figure 36.6, the default TX idles high and generates a low pulse when a 0 is transmitted. The polarity of TX may be inverted by setting the transmitter invert enable (TINVEN) bit, causing TX to idle low and generate a high pulse when sending a 0.

36.10.3.2. IrDA Demodulation

Setting the receiver IrDA enable (RIRDAEN) to 1 places the receiver in IrDA mode. In this mode, the receiver performs a simple RZ-to-NRZ (return-to-zero to non-return-to-zero) demodulation on the RX signal.

36.10.4. Smartcard Mode

The USART module includes hardware support for Smartcard interfaces as both a transmitter and a receiver.

36.10.4.1. Smartcard Transmitter

The transmitter Smartcard mode is enabled by setting the transmitter Smartcard parity response enable (TSCEN = 1) and consists of the ability to capture the ACK or NACK response from a Smartcard during the stop bit periods. Generally, Smartcard communications will also use multi-master and half-duplex modes with the TX and RX pins tied together externally.

Figure 36.7 illustrates the signaling on the TX pin in Smartcard mode.

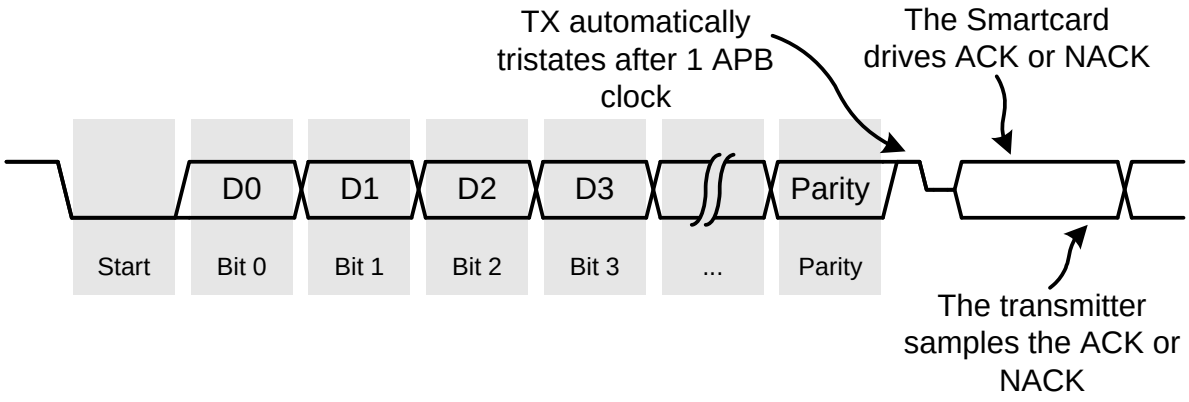


Figure 36.7. Smartcard Transmit Signalling

The transmitter should be configured to transmit 2 stop bits (or 1.5 stop bits if inter-packet delays are enabled) in Smartcard mode. The transmitter will tristate the TX pin one APB clock cycle after the start of the first stop bit. The Smartcard is expected to drive the TX pin low starting at the mid-point of the first stop bit if a receive error occurred. The transmitter captures the state of the TX pin at the end of the first stop bit period and sets the Smartcard parity error interrupt (TSCERI) flag accordingly. An interrupt is generated if the error flag is set and the transmit error interrupt is enabled (TERIEN = 1). The transmitter will re-enable its TX pin driver at the end of the stop bits except when operating in multi-master mode.

36.10.4.2. Smartcard Receiver

The receiver Smartcard mode is enabled by setting receiver Smartcard parity response (RSCEN) to 1 and allows the receiver to return an ACK or NACK response to a Smartcard during the stop bit periods. Figure 36.8 shows signaling on the RX pin in Smartcard mode.

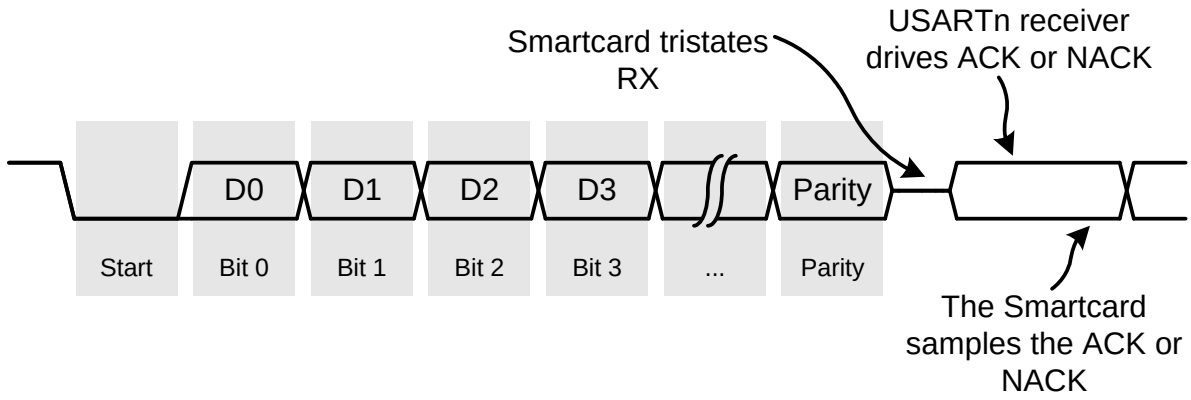


Figure 36.8. Smartcard Receive Signalling

When using Smartcard mode, firmware should configure the receiver to expect 2 stop bits. The receiver will drive RX low if a parity error occurs or high if no parity error occurs for one bit-time starting at the mid-point of the first stop bit. The Smartcard is expected to sample the state of RX at the end of the first stop bit.

36.10.5. LIN Support

The USART transmitter and receiver can be used with LIN (Local Interconnect Network) systems.

When using LIN, firmware should configure the transmitter to use 8 data bits, 1 start bit, 1 stop bit, and no parity bits.

The receiver auto-baud detection also supports the LIN SYNC byte (0x55).

36.10.6. Half Duplex Mode

The USART module supports applications in which half-duplex communications occur across a shared TX/RX signal. In half-duplex mode, the hardware automatically inhibits the receiver during transmit operations and the transmitter during receive operations.

The duplex mode bit (DUPLEXMD) enables half-duplex mode for the module. Firmware should also enable idle tristates (ITSEN = 1) to force the transmitter to tristate the TX signal when not transmitting. The RX and TX pins must be shorted externally in half-duplex mode.

36.10.7. Loop Back Support

The USART module supports several internal loop back options for testing purposes. The loop back modes are configured in the loop back mode (LBMD) field. Table 36.1 describes the different loop back options.

Table 36.1. Internal Loop Back Modes

LBMD value	Loop Back Mode
0	Disabled
1	Receive Loop Back
2	Transmit Loop Back
3	Full Loop Back

When loop back mode is disabled, the module operates normally.

In receive loop back mode, the receiver input path is disconnected from the RX signal and internally connected to the transmitter. Any data transmitted in this mode will be sent out on the TX signal and also received by the device. The physical RX pin tristates in this mode. TX, CTS, RTS, and UCLK are connected to the corresponding external pins, if the signals are enabled to connect to pins in the device's port configuration.

With transmit loop back, the transmitter output path is disconnected from the TX signal, and the RX input signal is internally looped back to the TX pin. Data received on the RX signal will be received by the device and also sent directly back out on the TX pin. RX, CTS, RTS, and UCLK are connected to the corresponding external pins, if enabled.

In full loop back mode, the transmitter output is internally routed back to the receiver input. Neither the transmitter nor receiver are connected to external device pins. The RX device pin is similarly looped back to the TX pin. Any data transmitted on TX will be sent directly back in on RX. The CTS input and RTS output pins are likewise looped back at both the transmitter and receiver and at the device pins, if enabled.

Figure 36.9 illustrates the internal connections for each loop back mode.

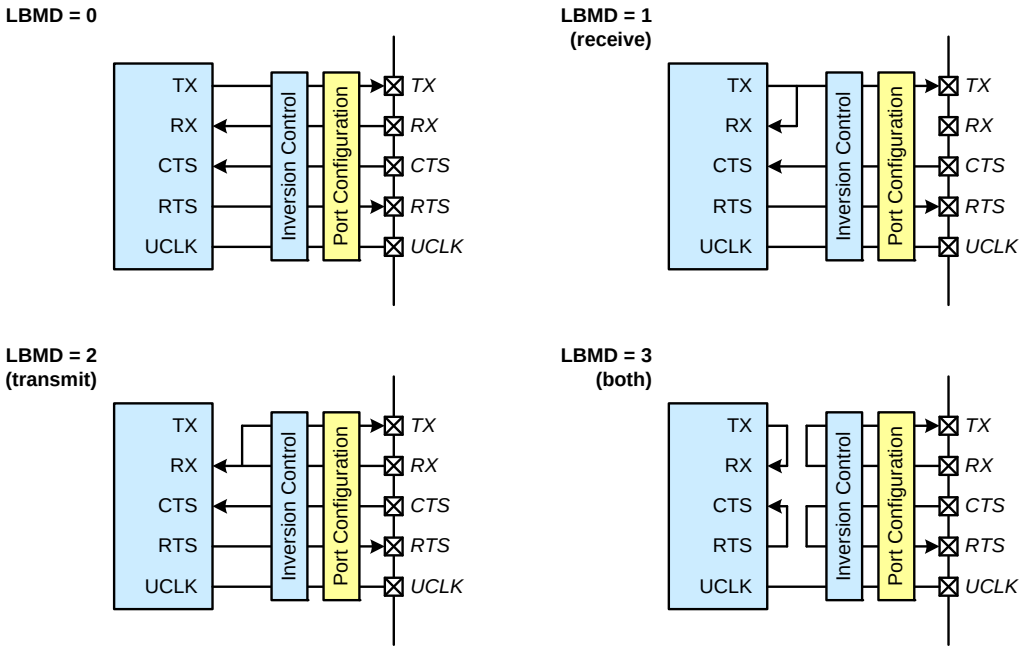


Figure 36.9. Internal Loop Back Connections

36.11. DMA Configuration and Usage

The USART module supports an independent DMA interface for the transmitter and receiver. The DMA operations should target the DATA register in non-incrementing mode. The number of bytes transferred per DMA read or write should coordinate with the transmit and receive FIFO thresholds (TFTH and RFTH) to ensure data is transferred as quickly as possible. For example, if transmitting 9-bit data, firmware can configure the TFTH field to cause a DMA data request when two FIFO entries are empty and the DMA to move words containing two 9-bit data values.

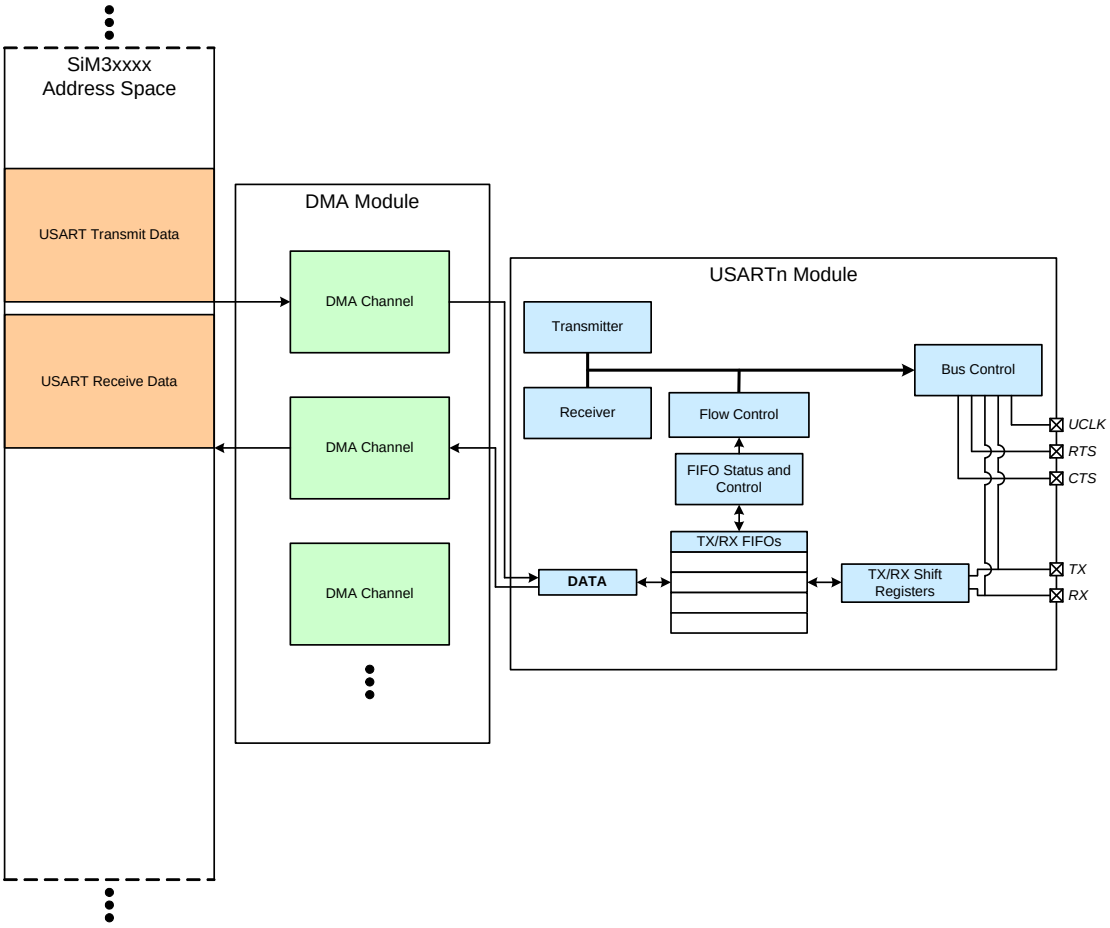


Figure 36.10. USART DMA Configuration

When the transmitter DMA enable (TDMAEN) bit is set and the number of empty entries in the transmit FIFO is equal to or greater than the setting in the transmit FIFO threshold (TFTH) field, the following two events occur:

1. The transmitter automatically requests a single DMA operation.
2. The transmitter sets the read-only transmit data request interrupt flag (TDREQI), which asserts an interrupt, if enabled (TDREQIEN = 1).

Firmware can enable DMA requests for the receiver by setting the RDMAEN bit to 1. When in DMA mode and the number of full entries in the receive FIFO is equal to or greater than the setting in the receive FIFO threshold (RFTH), the following two events occur:

1. The receiver automatically requests a single DMA operation. The receiver does not need to be enabled for this DMA request to occur.
2. The receiver sets the read-only receive data request interrupt flag (RDREQI), causing a receive data request interrupt, if enabled (RDREQIEN = 1).

The transmit and receive FIFO error flags are still set by hardware in DMA mode in the event of a FIFO error.

36.12. USART0 Registers

This section contains the detailed register descriptions for USART0 registers.

Register 36.1. USART0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TSYNCEN	TINVEN	TIRDAEN	TSCEN	Reserved	TDATLN			Reserved	TPARMD		TSTPMD		TSTPEN	TPAREN	TSTRTEN
Type	RW	RW	RW	RW	R	RW			R	RW		RW		RW	RW	RW
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RSYNCEN	RINVEN	RIRDAEN	RSCEN	Reserved	RDATLN			Reserved	RPARMD		RSTPMD		RSTPEN	RPAREN	RSTRTEN
Type	RW	RW	RW	RW	R	RW			R	RW		RW		RW	RW	RW
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1

Register ALL Access Address

USART0_CONFIG = 0x4000_0000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 36.2. USART0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	TSYNCEN	**Transmitter Synchronous Mode Enable.** 0: The transmitter operates in asynchronous mode. 1: The transmitter operates in synchronous mode.
30	TINVEN	**Transmitter Invert Enable.** 0: Do not invert the TX pin signals (the TX idle state is high). 1: Invert the TX pin signals (the TX idle state is low).
29	TIRDAEN	**Transmitter IrDA Enable.** 0: Disable IrDA transmit mode. 1: Enable IrDA transmit mode.
28	TSCEN	**Transmitter Smartcard Parity Response Enable.** 0: The transmitter does not check for a Smartcard parity error response. 1: The transmitter checks for a Smartcard parity error response.
27	Reserved	Must write reset value.

Table 36.2. USART0_CONFIG Register Bit Descriptions

Bit	Name	Function
26:24	TDATLN	**Transmitter Data Length.** Select the number of data bits sent during transmission. 000: 5 bits. 001: 6 bits. 010: 7 bits. 011: 8 bits. 100: 9 bits. The 9th bit is taken from the FIFO data (normal mode). 101: 9 bits. The 9th bit is set by the value of TBIT (fixed mode). 110-111: Reserved.
23	Reserved	Must write reset value.
22:21	TPARMD	**Transmitter Parity Mode.** This field selects the type of parity sent during transmissions. 00: Odd Parity. 01: Even Parity. 10: Set (Parity = 1). 11: Clear (Parity = 0).
20:19	TSTPMD	**Transmitter Stop Mode.** This bit selects the transmitted stop bit length. 00: 0.5 stop bit. 01: 1 stop bit. 10: 1.5 stop bits. 11: 2 stop bits.
18	TSTPEN	**Transmitter Stop Enable.** 0: Do not send stop bits during transmissions. 1: Send stop bits during transmissions.
17	TPAREN	**Transmitter Parity Enable.** 0: Do not send a parity bit during transmissions. 1: Send a parity bit during transmissions.
16	TSTRTEN	**Transmitter Start Enable.** 0: Do not generate a start bit during transmissions. 1: Generate a start bit during transmissions.
15	RSYNCEN	**Receiver Synchronous Mode Enable.** 0: The receiver operates in asynchronous mode. 1: The receiver operates in synchronous mode.
14	RINVEN	**Receiver Invert Enable.** 0: Do not invert the RX pin signals (the RX idle state is high). 1: Invert the RX pin signals (the RX idle state is low).
13	RIRDAEN	**Receiver IrDA Enable.** 0: The receiver does not operate in IrDA mode. 1: The receiver operates in IrDA mode.

Table 36.2. USART0_CONFIG Register Bit Descriptions

Bit	Name	Function
12	RSCEN	**Receiver Smartcard Parity Response Enable.** 0: The receiver does not send a Smartcard parity error response. 1: The receiver sends a Smartcard parity response.
11	Reserved	Must write reset value.
10:8	RDATLN	**Receiver Data Length.** Select the expected length of received data. 000: 5 bits. 001: 6 bits. 010: 7 bits. 011: 8 bits. 100: 9 bits. The 9th bit is stored in the FIFO (normal mode). 101: 9 bits. The 9th bit is not stored in the FIFO (fixed mode). This mode is used when the 9th bit is only used for match operations (see MATMD). 110-111: Reserved.
7	Reserved	Must write reset value.
6:5	RPARMD	**Receiver Parity Mode.** This field selects the type of parity expected during reception. 00: Odd Parity. 01: Even Parity. 10: Set (Parity = 1). 11: Clear (Parity = 0).
4:3	RSTPMD	**Receiver Stop Mode.** Select the number of stop bits expected during reception. 00: 0.5 stop bit. 01: 1 stop bit. 10: 1.5 stop bits. 11: 2 stop bits.
2	RSTPEN	**Receiver Stop Enable.** 0: Do not expect stop bits during receptions. 1: Expect stop bits during receptions.
1	RPAREN	**Receiver Parity Enable.** 0: Do not expect a parity bit during receptions. 1: Expect a parity bit during receptions.
0	RSTRTEN	**Receiver Start Enable.** 0: Do not expect a start bit during receptions. 1: Expect a start bit during receptions.

Register 36.2. USART0_MODE: Module Mode Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	OPMD	ITSEN	CLKESEL	CLKIDLE	DUPLEXMD	Reserved			ISTCLK	STRTSTCLK	STPSTCLK	Reserved	LBMD		Reserved	DBGMD
Type	RW	RW	RW	RW	RW	R			RW	RW	RW	R	RW		R	RW
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															
Type	R		RW		R	RW			R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

USART0_MODE = 0x4000_0010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 36.3. USART0_MODE Register Bit Descriptions

Bit	Name	Function
31	OPMD	**Operational Mode.** Select between master and slave operation in synchronous mode. This bit has no effect in asynchronous mode. 0: The USART operates as a slave. 1: The USART operates as a master.
30	ITSEN	**Idle TX/UCLK Tristate Enable.** 0: The TX and UCLK (if in synchronous master mode) pins are always an output in this mode, even when idle. 1: If ISTCLK is cleared to 0, the TX pin is tristated when idle. The UCLK pin will also be tristated when idle if in synchronous master mode.
29	CLKESEL	**Clock Edge Select.** This bit selects the edge control for the clock in synchronous mode. 0: The clock falls in the middle of each bit. 1: The clock rises in the middle of each bit.
28	CLKIDLE	**Clock Idle State.** Select the idle state for the clock in synchronous mode. 0: The synchronous clock is low when idle. 1: The synchronous clock is high when idle.

Table 36.3. USART0_MODE Register Bit Descriptions

Bit	Name	Function
27	DUPLEXMD	**Duplex Mode.** If this bit is set to 1, the ITSEN bit must also be set to 1 to tristate the TX signal during receive operations. 0: Full-duplex mode. The transmitter and receiver can operate simultaneously. 1: Half-duplex mode. The transmitter automatically inhibits when the receiver is active and the receiver automatically inhibits when the transmitter is active.
26:24	Reserved	Must write reset value.
23	ISTCLK	**Idle Clock Control.** Set whether or not the master generates clocks between transmissions in synchronous mode. If this bit is 1 when ITSEN is 1, the UCLK signal will not tristate when idle. 0: When the USART is a clock master, the clock is held idle between transmissions. 1: When the USART is a clock master, the clock is generated between transmissions or receptions.
22	STRTSTCLK	**Start State Clock Control.** Set whether or not the master generates clocks during the start bit in synchronous mode. 0: When the USART is a clock master, the clock is held idle during a start bit. 1: When the USART is a clock master, the clock is generated during a start bit.
21	STPSTCLK	**Stop State Clock Control.** Set whether or not the master generates clocks during the stop bit in synchronous mode. 0: When the USART is a clock master, the clock is not generated during stop bits. 1: When the USART is a clock master, the clock is generated during stop bits.
20	Reserved	Must write reset value.
19:18	LBMD	**Loop Back Mode.** Select from different internal loop-back options for diagnostic and debug purposes. 00: Loop back is disabled and the TX and RX signals are connected to the corresponding external pins. 01: Receive loop back. The receiver input path is disconnected from the RX pin and internally connected to the transmitter. Data transmitted will be sent out on TX and also received by the device. 10: Transmit loop back. The transmitter output path is disconnected from the TX pin and the RX input pin is internally looped back out to the TX pin. Data received at RX will be received by the device and also sent directly back out on TX. 11: Full loop back. Internally, the transmitter output is routed back to the receiver input. Neither the transmitter nor receiver are connected to external device pins. The device pin RX is looped back to TX in a similar fashion. Data transmitted on TX will be sent directly back in on RX.
17	Reserved	Must write reset value.

Table 36.3. USART0_MODE Register Bit Descriptions

Bit	Name	Function
16	DBGMD	**USART Debug Mode.** 0: The USART module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the USART module to halt. Any active transmissions and receptions will complete first.
15:0	Reserved	Must write reset value.

Register 36.3. USART0_FLOWCN: Flow Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		TIRDAPW		Reserved				CTSEN	Reserved	CTSINVEN	Reserved		UCLK	TX	CTS
Type	R		RW		R				RW	R	RW	R		RW	RW	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			TXOEN	Reserved				RTSEN	RTSTH	RTSINVEN	Reserved			RX	RTS
Type	R			RW	R				RW	RW	RW	R			R	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1

Register ALL Access Address

USART0_FLOWCN = 0x4000_0020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 36.4. USART0_FLOWCN Register Bit Descriptions

Bit	Name	Function
31:30	Reserved	Must write reset value.
29:28	TIRDAPW	**Transmit IrDA Pulse Width.** This field sets the IrDA pulse width as a fraction of the bit period. 00: The IrDA pulse width is 1/16th of a bit period. 01: The IrDA pulse width is 1/8th of a bit period. 10: The IrDA pulse width is 3/16th of a bit period. 11: The IrDA pulse width is 1/4th of a bit period.
27:24	Reserved	Must write reset value.
23	CTSEN	**CTS Enable.** 0: The CTS pin state does not affect transmissions. 1: Transmissions will begin only if the CTS pin (after optional inversion) is low.
22	Reserved	Must write reset value.
21	CTSINVEN	**CTS Invert Enable.** 0: The USART does not invert CTS. 1: The USART inverts CTS.
20:19	Reserved	Must write reset value.

Table 36.4. USART0_FLOWCN Register Bit Descriptions

Bit	Name	Function
18	UCLK	**UCLK State.** This bit may be written when both the transmitter and receiver are disabled (TEN=REN=0) and in asynchronous mode (TSYNCEN=RSYCNEN=0). Note that the written value will not affect the UCLK output unless the USART is configured as a Master (OPMD=1) and idle tri-state is disabled (ITSEN=0). 0: The UCLK pin is low. 1: The UCLK pin is high.
17	TX	**TX State.** Firmware can write this bit to change the TX state only when the transmitter is disabled (TEN = 0), the TX output is enabled (TXOEN =1), and idle tri-state is disabled (ITSEN=0). 0: The TX pin (before optional inversion) is low. 1: The TX pin (before optional inversion) is high.
16	CTS	**CTS State.** 0: Indicates the CTS pin state (after optional inversion) is low. 1: Indicates the CTS pin state (after optional inversion) is high.
15:13	Reserved	Must write reset value.
12	TXOEN	**TX Output Enable.** 0: The pin assigned to TX is tri-stated, regardless of other settings. 1: The pin assigned to TX is controlled by the USART.
11:8	Reserved	Must write reset value.
7	RTSEN	**RTS Enable.** 0: The RTS state is not changed by hardware. The RTS bit can be written only when hardware RTS is disabled (RTSEN = 0). 1: Hardware sets RTS when the receive FIFO is at or above the threshold set by RTSTH and clears RTS otherwise.
6	RTSTH	**RTS Threshold Control.** 0: RTS is de-asserted when the receive FIFO and shift register are full and no more incoming data can be stored. 1: RTS is de-asserted when the receive FIFO and shift register are nearly full and only one more data can be received.
5	RTSINVEN	**RTS Invert Enable.** 0: The USART does not invert the RTS signal before driving the pin. 1: The USART inverts the RTS signal driving the pin.
4:2	Reserved	Must write reset value.
1	RX	**RX Pin Status.** 0: RX pin (after optional inversion) is low. 1: RX pin (after optional inversion) is high.

Table 36.4. USART0_FLOWCN Register Bit Descriptions

Bit	Name	Function
0	RTS	**RTS State.** This bit is writeable only when RTSEN is cleared to 0. 0: RTS pin (before optional inversion) is driven low. 1: RTS pin (before optional inversion) is driven high.

Register 36.4. USART0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TEN	TINH	Reserved	TBIT	TBUSYF	Reserved			TCPTIEN	TDREQIEN	TERIEN	TCPTTH	TCPTI	TDREQI	TUREI	TSCERI
Type	RW	RW	R	RW	R	R			RW	RW	RW	RW	RW	R	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	REN	RINH	ROSEN	RBIT	RBUSYF	RABDEN	MATMD		Reserved	RDREQIEN	RERIEN	Reserved	RDREQI	ROREI	RPARERI	RFRMERI
Type	RW	RW	RW	RW	R	RW	RW		R	RW	RW	R	R	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

USART0_CONTROL = 0x4000_0030

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 36.5. USART0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	TEN	**Transmitter Enable.** 0: Disable the transmitter. When cleared, the transmitter immediately aborts any active transmission. Clearing this bit does not automatically flush the transmit FIFO. 1: Enable the transmitter. The transmitter will initiate a transmission when data becomes available in the transmit FIFO.
30	TINH	**Transmit Inhibit.** 0: The transmitter operates normally. 1: Transmissions are inhibited. The transmitter will stall after any current transmission is complete.
29	Reserved	Must write reset value.
28	TBIT	**Last Transmit Bit.** This bit is used to set the 9th bit during each FIFO write when TDATLN is set to 5. The TBIT value is stored in the FIFO during each FIFO write; the TBIT value at the time of transmission is not used. If TDATLN is not set to 5, the written bit from the core interface is used. This allows the DMA or the core to pack 9-bit outgoing data into bytes when the outgoing 9th bit is constant.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 36.5. USART0_CONTROL Register Bit Descriptions

Bit	Name	Function
27	TBUSYF	**Transmitter Busy Flag.** 0: The USART transmitter is idle. 1: The USART transmitter is active and transmitting.
26:24	Reserved	Must write reset value.
23	TCPTIEN	**Transmit Complete Interrupt Enable.** 0: Disable the transmit complete interrupt. 1: Enable the transmit complete interrupt. A transmit interrupt is generated when TCPTI is set to 1.
22	TDREQIEN	**Transmit Data Request Interrupt Enable.** 0: Disable the transmit data request interrupt. 1: Enable the transmit data request interrupt. A transmit interrupt is asserted when TDREQI is set to 1.
21	TERIEN	**Transmit Error Interrupt Enable.** 0: Disable the transmit error interrupt. 1: Enable the transmit error interrupt. A transmit interrupt is generated when TUREI or TSCERI is set to 1.
20	TCPTTH	**Transmit Complete Threshold.** 0: The TCPTI flag is set after each data transmission. 1: The TCPTI flag is set after transmission of the last available data.
19	TCPTI	**Transmit Complete Interrupt Flag.** This bit is set by hardware if a byte is transmitted (TCCPTH = 0) or if the last available byte is transmitted (TCPTTH = 1). This bit must be cleared by firmware.
18	TDREQI	**Transmit Data Request Interrupt Flag.** 0: The transmitter is not requesting more FIFO data. 1: The transmitter is requesting more FIFO data.
17	TUREI	**Transmit Underrun Error Interrupt Flag.** Hardware sets this bit to 1 when a transmit FIFO underrun occurs. This bit must be cleared by firmware.
16	TSCERI	**Smartcard Parity Error Interrupt Flag.** This bit is set by hardware when a Smartcard parity error occurs. This bit must be cleared by firmware.
15	REN	**Receiver Enable.** This bit enables the USART receiver for multiple transactions. 0: Disable the receiver. The receiver can receive one data transaction only if ROSEN is set. 1: Enable the receiver.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 36.5. USART0_CONTROL Register Bit Descriptions

Bit	Name	Function
14	RINH	**Receiver Inhibit.** 0: The receiver operates normally. 1: RTS is immediately asserted when RINH is set. The receiver will complete any ongoing reception, but ignore all traffic after that.
13	ROSEN	**Receiver One-Shot Enable.** 0: Disable one-shot receive mode. 1: Enable one-shot receive mode.
12	RBIT	**Last Receive Bit.** This bit is used according to the match mode (MATMD) setting.
11	RBUSYF	**Receiver Busy Flag.** 0: The USART receiver is idle. 1: The USART receiver is receiving data.
10	RABDEN	**Receiver Auto-Baud Enable.** 0: Disable receiver auto-baud. 1: Enable receiver auto-baud.
9:8	MATMD	**Match Mode.** The hardware automatically switches from MCE mode to Frame mode when a MCE match occurs. 00: Disable the match function. 01: (MCE) Data whose last data bit equals RBIT is accepted and stored. 10: (Frame) A framing error is asserted if the last received data bit matches RBIT. 11: (Store) Store the last incoming data bit in RBIT. This mode can be used incon-junction with the RDATLN setting.
7	Reserved	Must write reset value.
6	RDREQIEN	**Receive Data Request Interrupt Enable.** 0: Disable the read data request interrupt. 1: Enable the read data request interrupt. A receive interrupt is generated when RDREQI is set to 1.
5	RERIEN	**Receive Error Interrupt Enable.** 0: Disable the receive error interrupt. 1: Enable the receive error interrupt. A receive interrupt is asserted when ROREI, RFRMERI, or RPARERI is set to 1.
4	Reserved	Must write reset value.
3	RDREQI	**Receive Data Request Interrupt Flag.** 0: Fewer than RFTH FIFO entries are filled with data. 1: At least RFTH FIFO entries are filled with data.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 36.5. USART0_CONTROL Register Bit Descriptions

Bit	Name	Function
2	ROREI	**Receive Overrun Error Interrupt Flag.** This bit is set to 1 by hardware when a receive overrun error occurs. This bit must be cleared by firmware.
1	RPARERI	**Receive Parity Error Interrupt Flag.** This bit is set to 1 by hardware when the receiver encounters a parity error. This bit must be cleared by firmware.
0	RFRMERI	**Receive Frame Error Interrupt Flag.** Hardware sets this bit to 1 when a receive frame error occurs. There are two sources for this error: when an expected whole stop bit is low, or when a frame match-mode fails. This bit must be cleared by firmware.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 36.5. USART0_IPDELAY: Inter-Packet Delay

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								IPDELAY							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

USART0_IPDELAY = 0x4000_0040

Table 36.6. USART0_IPDELAY Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:16	IPDELAY	**Inter-Packet Delay.** This field configures the transmitter to delay between transmissions by the specified number of bit times. A value of 0 means no delay is added, and a value of 5 means 5 bit times are added. Bit times are configured in the TBAUD register. If in synchronous slave mode, UCLK must be running to use the inter-packet delay feature.
15:0	Reserved	Must write reset value.

Register 36.6. USART0_BAUDRATE: Transmit and Receive Baud Rate

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TBAUD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RBAUD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
USART0_BAUDRATE = 0x4000_0050

Table 36.7. USART0_BAUDRATE Register Bit Descriptions

Bit	Name	Function
31:16	TBAUD	**Transmitter Baud Rate Control.** This field sets the transmitter baud rate according to the equation: $\text{Baud Rate} = \frac{F_{APB}}{N \times (TBAUD + 1)}$ N = 2 if TIRDAEN = 0, and N = 16 if TIRDAEN = 1. Note that in synchronous mode, the receiver baud rate must not exceed APBCLK / 16.
15:0	RBAUD	**Receiver Baud Rate Control.** This field sets the receiver baud rate according to the equation: $\text{Baud Rate} = \frac{F_{APB}}{N \times (RBAUD + 1)}$ N = 2 if RIRDAEN = 0, and N = 16 if RIRDAEN = 1. Note that in synchronous mode, the receiver baud rate must not exceed APBCLK / 16.

Register 36.7. USART0_FIFOCN: FIFO Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved					TSRFULLF	TFERI	TFIFOFL	TDMAEN	Reserved	TFTH		Reserved	TCNT		
Type	R					R	RW	RW	RW	R	RW		R	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					RSRFULLF	RFERI	RFIFOFL	RDMAEN	Reserved	RFTH		Reserved	RCNT		
Type	R					R	RW	RW	RW	R	RW		R	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

USART0_FIFOCN = 0x4000_0060

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 36.8. USART0_FIFOCN Register Bit Descriptions

Bit	Name	Function
31:27	Reserved	Must write reset value.
26	TSRFULLF	**Transmit Shift Register Full Flag.** This bit indicates when the transmitter shift register contains data. The bit is set when hardware loads the data from the FIFO to the transmit shift register and is cleared when the transmitter completely transmits the data.
25	TFERI	**Transmit FIFO Error Interrupt Flag.** This bit is set when an illegal FIFO write is detected, such as a non right-justified write or a write that contains more data than will fit in the empty FIFO entries. This bit must be cleared by firmware. 0: A transmit FIFO error has not occurred since TFERI was last cleared. 1: A transmit FIFO error occurred.
24	TFIFOFL	**Transmit FIFO Flush.** Setting this bit to 1 flushes the transmit FIFO. If data is pending in the transmit shift register but a transmit has not begun, the shift register is also flushed. This bit always reads as 0.

Table 36.8. USART0_FIFOCN Register Bit Descriptions

Bit	Name	Function
23	TDMAEN	**Transmitter DMA Enable.** Enable the transmit FIFO DMA request. This request is generated according to the TFTH setting. 0: Disable transmit FIFO DMA requests. 1: Enable transmit FIFO DMA requests.
22	Reserved	Must write reset value.
21:20	TFTH	**Transmit FIFO Threshold.** This field sets the FIFO threshold at which a DMA transfer request or a TDREQI interrupt is asserted. 00: A DMA request or transmit data request interrupt (TDREQI) is asserted when >= 1 FIFO entry is empty. 01: A DMA request or transmit data request interrupt (TDREQI) is asserted when >= 2 FIFO entries are empty. 10: A DMA request or transmit data request interrupt (TDREQI) is asserted when >= 3 FIFO entries are empty. 11: A DMA request or transmit data request interrupt (TDREQI) is asserted when >= 4 FIFO entries are empty.
19	Reserved	Must write reset value.
18:16	TCNT	**Transmit FIFO Count.** This field indicates the number of entries in the transmit FIFO.
15:11	Reserved	Must write reset value.
10	RSRFULLF	**Receive Shift Register Full Flag.** This bit indicates when the receiver shift register contains data and remains high until the data is moved to the FIFO or is flushed. The flag is set as soon as incoming data is completely received and cleared when the data is transferred to the FIFO.
9	RFERI	**Receive FIFO Error Interrupt Flag.** This bit is set when hardware detects an illegal FIFO read, such as a non right-justified read or a read that demands more data than is available in the FIFO. 0: A receive FIFO error has not occurred since RFERI was last cleared. 1: A receive FIFO error occurred.
8	RFIFOFL	**Receive FIFO Flush.** Setting this bit to 1 flushes the receive FIFO and any completed data in the receive shift register. This bit always reads as 0.
7	RDMAEN	**Receiver DMA Enable.** Enable the receive FIFO DMA request. This request is generated according to the RFTH setting. 0: Disable receive FIFO DMA requests. 1: Enable receive FIFO DMA requests.
6	Reserved	Must write reset value.

Table 36.8. USART0_FIFOCN Register Bit Descriptions

Bit	Name	Function
5:4	RFTH	**Receive FIFO Threshold.** This is the threshold at which a DMA transfer or a RDREQI interrupt is asserted. 00: A DMA request or read data request interrupt (RDREQI) is asserted when >= 1 FIFO entry is full. 01: A DMA request or read data request interrupt (RDREQI) is asserted when >= 2 FIFO entries are full. 10: A DMA request or read data request interrupt (RDREQI) is asserted when >= 3 FIFO entries are full. 11: A DMA request or read data request interrupt (RDREQI) is asserted when >= 4 FIFO entries are full.
3	Reserved	Must write reset value.
2:0	RCNT	**Receive FIFO Count.** This field indicates the number of entries in the receive FIFO.

Register 36.8. USART0_DATA: FIFO Input/Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

USART0_DATA = 0x4000_0070

Table 36.9. USART0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**FIFO Data.** The 32-bit DATA register is a single port into the transmit and receive FIFOs. Reads and writes should always be right-justified. Byte-wide reads and writes should always access DATA[7:0], half-word reads and writes should always access DATA[15:0], and word reads and writes should always access DATA[31:0]. Writes to the DATA register push data into the transmit FIFO. Reads from the DATA register pop data from the receive FIFO. Multiple FIFO entries can be pushed or popped in a single write or read. For example, if the transmit bit-length is less than 9 bits, writing a whole word (4 bytes) to the DATA field will write four entries in the FIFO where each byte is a single entry. However, if the FIFO doesn't have room for 4 entries, the write is completely ignored and the TFERI error flag is set. Similarly, a half-word write will push 2 entries to the FIFO if the data length is < 9. If the data length is 9, each half-word is a single entry. When writing or reading multiple bytes from the FIFO, the least significant byte is written to or read from the FIFO first.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

36.13. USART0 Register Memory Map

Table 36.10. USART0 Memory Map

USART0_CONTROL	USART0_FLOWCN	USART0_MODE	USART0_CONFIG	Register Name
0x4000_0030	0x4000_0020	0x4000_0010	0x4000_0000	ALL Address
ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR	Access Methods
TEN	Reserved	OPMD	TSYNCEN	Bit 31
TINH		ITSEN	TINVEN	Bit 30
Reserved	TIRDAPW	CLKESEL	TIRDAEN	Bit 29
TBIT		CLKIDLE	TSCEN	Bit 28
TBUSYF	Reserved	DUPLEXMD	Reserved	Bit 27
Reserved		Reserved	TDATLN	Bit 26
				Bit 25
				Bit 24
TCPTIEN	CTSEN	ISTCLK	Reserved	Bit 23
TDREQIEN	Reserved	STRTSTCLK	TPARMD	Bit 22
TERIEN	CTSINVEN	STPSTCLK		Bit 21
TCPTTH	Reserved	Reserved	TSTPMD	Bit 20
TCPTI		LBMD		Bit 19
TDREQI	UCLK		TSTPEN	Bit 18
TUREI	TX	Reserved	TPAREN	Bit 17
TSCERI	CTS	DBGMD	TSTRTEN	Bit 16
REN	Reserved	Reserved	RSYNCEN	Bit 15
RINH			RINVEN	Bit 14
ROSEN			RIRDAEN	Bit 13
RBIT	TXOEN		RSCEN	Bit 12
RBUSYF	Reserved		Reserved	Bit 11
RABDEN			RDATLN	Bit 10
MATMD				Bit 9
				Bit 8
Reserved	RTSEN		Reserved	Bit 7
RDREQIEN	RTSTH		RPARMD	Bit 6
RERIEN	RTSINVEN			Bit 5
Reserved	Reserved		RSTPMD	Bit 4
RDREQI				Bit 3
ROREI			RSTPEN	Bit 2
RPARERI	RX		RPAREN	Bit 1
RFRMERI	RTS		RSTRTEN	Bit 0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 36.10. USART0 Memory Map

Register Name	USART0_DATA	USART0_FIFOCN	USART0_BAUDRATE	USART0_IPDELAY
ALL Address	0x4000_0070	0x4000_0060	0x4000_0050	0x4000_0040
Access Methods	ALL	ALL | SET | CLR	ALL	ALL
Bit 31	DATA	Reserved	TBAUD	Reserved
Bit 30		TSRFULLF		
Bit 29		TFERI		
Bit 28		TFIFOFL		
Bit 27		TDMAEN		
Bit 26		Reserved		
Bit 25		TFTH		
Bit 24		Reserved		
Bit 23		TCNT		IPDELAY
Bit 22				
Bit 21				
Bit 20				
Bit 19				
Bit 18				
Bit 17				
Bit 16				
Bit 15		Reserved	RBAUD	Reserved
Bit 14		RSRFULLF		
Bit 13		RFERI		
Bit 12		RFIFOFL		
Bit 11		RDMAEN		
Bit 10		Reserved		
Bit 9		RFTH		
Bit 8		Reserved		
Bit 7		RCNT		
Bit 6				
Bit 5				
Bit 4				
Bit 3				
Bit 2				
Bit 1				
Bit 0				

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

37. Universal Asynchronous Receiver/Transmitter (UART0)

This section describes the UART module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the UART block, which is used by UART0 on all device families covered in this document.

37.1. UART Features

The UART module includes the following features:

- Independent transmitter and receiver configurations with separate 16-bit baud-rate generators.
- Ability to operate at 9600, 4800, 2400 or 1200 baud from RTC0TCLK running at 32.768 kHz.
- Power Mode 8 (PM8) Wake source.
- Asynchronous transmissions and receptions.
- Up to 5 Mbaud (TX or RX).
- Internal transmit and receive FIFOs with flush capability and support for byte, half-word, and word reads and writes.
- Data bit lengths from 5 to 9 bits.
- Programmable inter-packet transmit delays.
- Auto-baud detection with support for the LIN SYNC byte.
- Automatic parity generation (with enable).
- Automatic start and stop generation.
- Independent inversion correction for TX and RX signals.
- IrDA modulation and demodulation with programmable pulse widths.
- Parity error, frame error, overrun, and underrun detection.
- Multi-master and half-duplex support.
- Multiple loop-back modes supported.

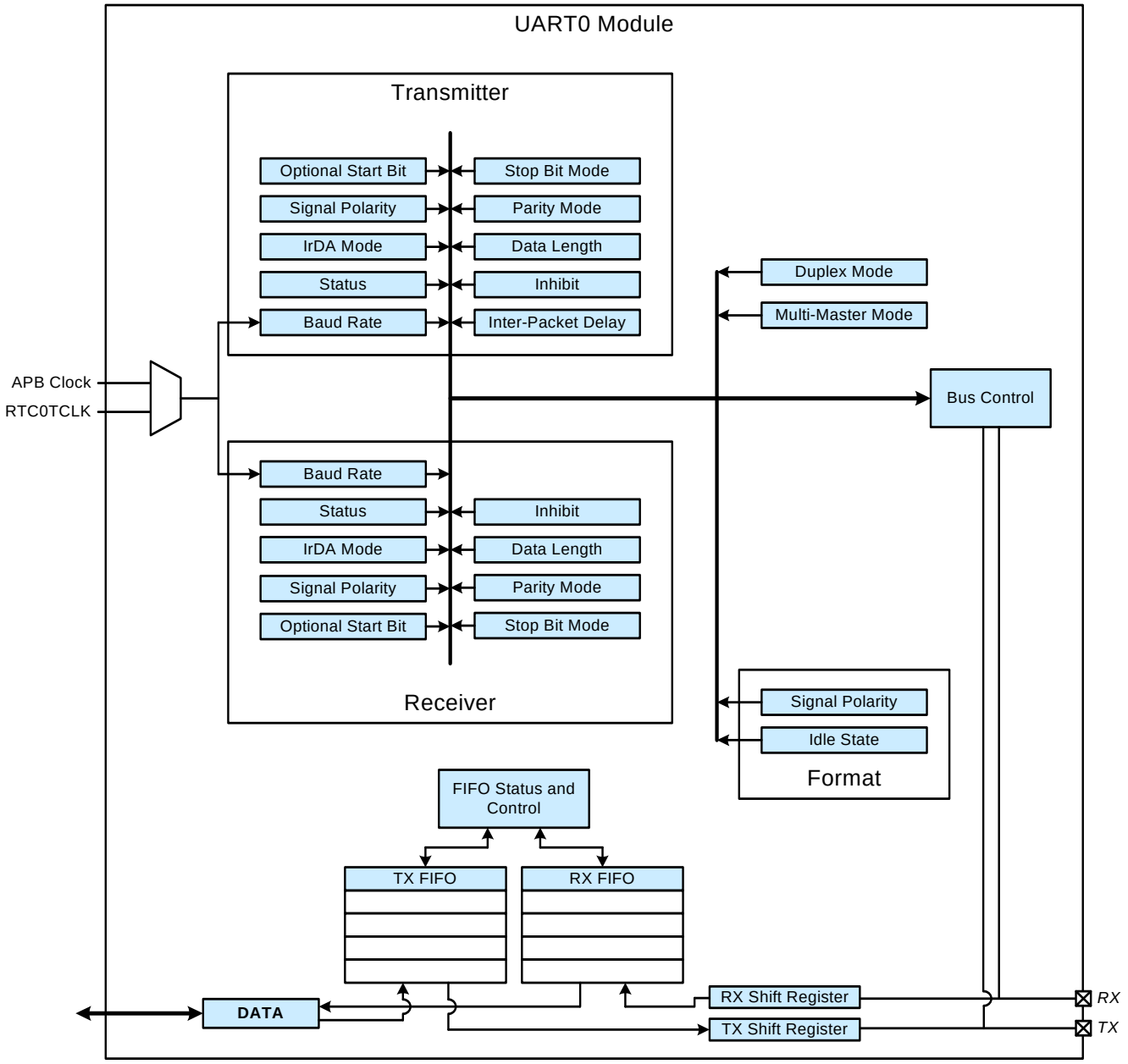


Figure 37.1. UART Block Diagram

37.2. Pin Assignment

The UART0 peripheral is available only on dedicated port I/O pins. When the UART0 peripheral is enabled, it is automatically mapped to the appropriate pins, as shown in Table 37.1.

Both the UART0_TX and UART0_RX pins should be configured as digital inputs in the Port Bank standard configuration registers.

Table 37.1. UART0 Pin Mapping

UART0 Signal	SiM3L1x7 Pin Name	SiM3L1x6 Pin Name	SiM3L1x4 Pin Name
UART0_TX	PB1.2	PB1.2	PB0.7
UART0_RX	PB1.3	PB1.3	PB0.8

37.3. UART Clocking

The UART module has two distinct clock sources available: the APB clock and the RTC timer clock (RTC0TCLK). The APB clock option allows the UART to operate at higher baud rates while the RTC timer clock option allows the UART to support very low power operation and be used as a wake source from power mode PM8. When operating the UART from the RTC0TCLK source there are two possibilities. Either the APB clock is also run from the RTC0TCLK, or the APB clock is run at a higher rate while only the UART operates from the RTC0TCLK.

37.3.1. Standard Mode (APB Clock Source)

To operate the UART in standard mode from an APB clock source, the RTCBDMD and RTCCKMD bits in the MODE register should be cleared to 0. In standard operation mode the UART retains all features except for the ability to operate in power mode PM8. The CLKDIV register should be configured so that the UART module clock is no faster than ~12 MHz as shown in Table 37.2.

Table 37.2. UART0 CLKDIV Configuration

UART0 Module Clock Source	UART0 Module Clock Frequency	UART0 CLKDIV Value
APB Clock	$\geq$ 25 MHz	0x10 (Divide by 4)
APB Clock	< 25 MHz	0x01 (Divide by 2)
RTC Clock	32.768 kHz	0x00 (Divide by 1)

37.3.2. RTC Timer Clock Modes (RTC0TCLK Source) - Low Power PM8 Operation

When running from a 32.768 kHz RTC0TCLK, four baud rates are possible: 9600, 4800, 2400 and 1200. These are selected using bits 1-0 of the RBAUD or TBAUD fields as detailed in the BAUDRATE register description. When using this special mode of the UART, it is possible to put the device into power mode PM8, and wake the device on UART traffic. IrDA and autobaud operation are not supported in this mode.

If the APB clock is configured to run from the RTC0TCLK, the RTCBDMD bit in the MODE register should be set to 1. Both the baud rate and the register access are derived from the RTC0TCLK source, allowing the register access to operate without re-synchronization between the clock domains.

If the APB clock is not configured to run from RTC0TCLK but it is desired to have the UART running from RTC0TCLK, the RTCCKMD bit in the MODE register should be set to 1. In this mode, the APB clock frequency must be at least four times the RTC0TCLK frequency to ensure proper synchronization between the two clock domains when accessing the UART registers.

To facilitate the lowest power operation possible, the UART implements some additional clocking features when running from the RTC0TCLK. By default, the UART baud clock will not run unless it is needed. If a transmit or receive operation is detected, the UART baud clock will become active. This operation may be overridden if desired by writing the FORCECLK bit in the MODE register to 1. This will force the UART baud clock to always be running.

The RXCLKSW and TXCLKSW bits in the MODE register enable automatic switching of the UART clock source between the RTC0TCLK and the APB clock source when an interrupt is generated. If automatic switching is enabled, the value in RBAUD or TBAUD should be compatible with both the APB clock and RTC0TCLK modes. For example, if the APB clock is 20 MHz and the desired baud rate is 9600, the RBAUD and TBAUD should be set to 0x411: in APB clock mode, 0x411 selects a baudrate of 9596 and in RTC0TCLK mode only the two least significant bits are used (i.e., 0x11) to select a baudrate of 9600. Additionally, the UART operations (receive or transmit) should be timed to allow for at least two idle bit periods when the interrupt will occur. Otherwise, the clock switch may cause an undesirable glitch in the UART timing.

37.4. Basic Data Format

The UART module data consists of four parts: start bit, data, parity bit, and stop bit.

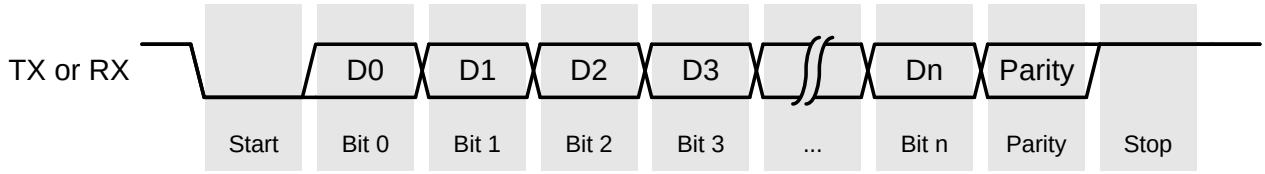


Figure 37.2. Basic Asynchronous UART Communication

Start bits are enabled using the transmitter start enable (TSTRTEN) and receiver start enable (RSTRTEN) bits. Disabling the start bits is not generally used for asynchronous operation.

The data length is variable and can be set to 5 to 9 bits using the transmitter data length (TDATLN) or receiver data length (RDATLN) bit fields.

The transmitter will transmit a parity bit when the transmitter parity enable (TPAREN) bit is set. The transmitter parity mode (TPARMD) field configures the parity for the transmit data as either odd, even, set (parity always set to 1), or clear (parity always cleared to 0).

Similarly, the receiver will expect to receive a parity bit when receiver parity is enabled (RPAREN = 1). The RPARMD field controls the receiver parity mode and sets it to odd, even, set (parity always expected to be set to 1), or clear (parity always expected to be cleared to 0).

The stop bits are also fully configurable using the transmitter stop enable (TSTPEN) and receiver stop enable (RSTPEN) bits. The transmitter stop mode (TSTPMD) and receiver stop mode (RSTPMD) fields configure the stop length to 0.5, 1, 1.5, or 2 bits. Transmitting partial stop bits (0.5 or 1.5) causes subsequent transmitted bits to slip by one-half of a bit-time. When operating in RTC timer clock mode, only the 1 or 2 stop bit options are available.

37.5. Baud Rate

The UART can operate as two independent, one-way channels since the transmitter and the receiver each have their own internal baud rate generators, FIFOs and shift registers. For example, the transmitter can operate at one baud rate while the receiver operates at a different baud rate.

In master mode, the transmitter baud rate control (TBAUD) bit field sets the transmitter baud rate, and the receiver baud rate control (RBAUD) bit field sets the receiver baud rate. The equations in the TBAUD and RBAUD field descriptions determine the transmitter or receiver baud rate. Note that the derivation of baud rate is different, depending on whether the UART is clocked from the APB clock or the RTC0TCLK.

37.5.1. Receiver Auto-Baud Detection

The hardware receiver auto-baud detection automatically derives the appropriate RBAUD value from the received data. When receive auto-baud detection is enabled (RABDEN = 1), the hardware requires the first incoming data to be a value that guarantees an RX state transition between each bit: 0x55 when in non-IrDA mode and 0x00 when in IrDA mode. The receiver auto-baud detection supports either 8- or 9-bit data lengths with any combination of parity or stop bits.

Figure 37.3 shows the UART receive auto-baud detection input timing.

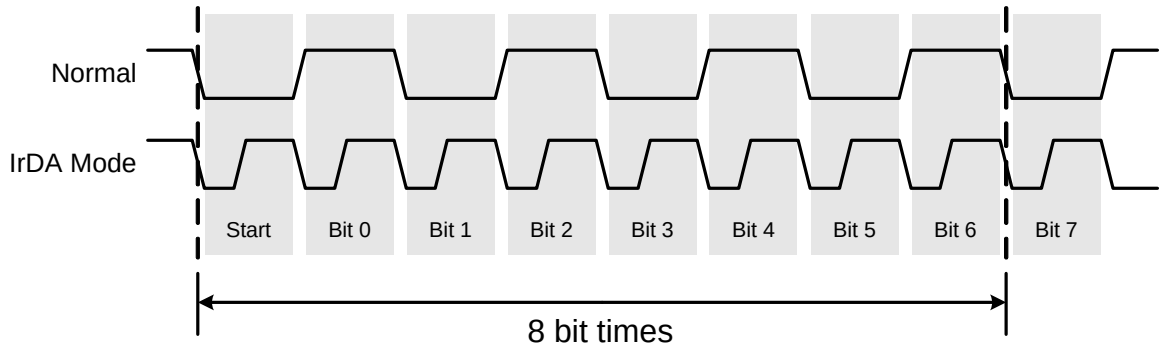


Figure 37.3. Receiver Auto-Baud Detection Input Timing

The receiver auto-baud detection calculates the appropriate value for RBAUD by counting the number of clock cycles between the leading edge of the start bit and the leading edge of the 8th data bit and dividing the clock cycle count by either 16 (when RIRDAEN = 0) or 128 (when RIRDAEN = 1). The hardware stores the calculated value to RBAUD and clears the receive auto-baud detection enable bit (RABDEN = 0). The new baud rate is used starting at the 8th bit of the incoming packet. The receiver will treat the received data (0x55 or 0x00) like any other received data and check it for parity errors and store it in the FIFO. If the first incoming byte causes the RBAUD value to overflow, indicating an auto-baud error, hardware will set the receive frame error interrupt (RFRMERI) flag regardless of the sampled state of the stop bits.

Auto-Baud detection is not supported when operating from the RTC0TCLK.

37.6. Interrupts

The UART module contains several interrupt sources that cause a vector to the UART interrupt. All of the interrupt flags may be masked by clearing a corresponding interrupt enable bit.

37.6.1. Transmit Interrupt Sources

The transmit data request interrupt (TDREQI) can be enabled by setting the TDREQIEN bit to 1. This flag indicates that the transmitter is requesting more FIFO data. The TDREQI flag automatically clears when the number of full entries in the transmit FIFO is more than the TFTH setting.

The transmit complete interrupt (TCPTI) flag indicates the following:

1. If TCPTTH = 0, this flag indicates a single transmit completed since TCPTI was last cleared.
2. If TCPTTH = 1, this flag indicates a transmit of the last available data in the FIFO completed since TCPTI was last cleared.

In either scenario, the hardware will generate an interrupt if the TCPTIEN bit is set to 1 when TCPTI is set.

Finally, the transmitter has a transmit FIFO error interrupt (TFERI) flag which is set whenever an illegal FIFO write (e.g., a write to a full FIFO) is detected. Note that TFERI does not have an interrupt enable/disable control bit, so if the TFERI flag is set an interrupt will always be generated (assuming the UART module interrupt is enabled).

37.6.2. Receive Interrupt Sources

The receive data request interrupt (RDREQI) flag indicates that at least the number of receive FIFO slots set by the receive FIFO threshold (RFTH) are full, and firmware can read data from the receive FIFO. This interrupt is enabled by setting the receive data request interrupt enable (RDREQIEN) to 1. This flag automatically clears when the number of filled entries in the receive FIFO drops below the RFTH setting.

The receiver has three error interrupt sources. The receive overrun error interrupt (ROREI) flag indicates when the receive FIFO and shift register is full, and the shift register has been erroneously overwritten by additional receive data. The receive parity error interrupt (RPARERI) indicates that an invalid parity bit has been received. Finally, the receive frame error interrupt (RFRMERI) indicates when an expected whole stop bit is low, when a frame match mode fails, or when an auto-baud error occurs. The receive error interrupt enable (RERIEN) bit enables these flags as interrupt sources.

Finally, the receiver has a receive FIFO error interrupt (RFERI) flag which is set whenever an illegal FIFO read (e.g, a read from an empty FIFO) is detected. Note that RFERI does not have an interrupt enable/disable control bit, so if the RFERI flag is set an interrupt will always be generated (assuming the UART module interrupt is enabled).

37.7. Inter-Packet Delay Generator

The transmitter supports a configurable delay between transmissions that may be used to limit the transmission rate in applications that do not support hardware flow control.

The hardware calculates the inter-packet delay using a counter operating at the current baud rate. As a result, the IPDELAY field represents multiples of the transmitter bit times. For example, setting the inter-packet delay (IPDELAY) field to 5 results in a delay equal to 5 bit times. Setting the IPDELAY field to 0 disables the inter-packet delay.

37.8. Debug Mode

Firmware can set the DBGMD bit to force the UART module to halt on a debug breakpoint. The module will complete any active transmissions and receptions before stopping. Clearing the DBGMD bit forces the UART module to continue operating while the core halts in debug mode. Note that when IPDELAY is set to zero, the UART may not halt until the FIFO is empty. If IPDELAY is non-zero, the UART will complete the current transfer but will not empty the FIFO.

37.9. Sending Data

To begin a transmit operation, firmware should set all the necessary transmitter configuration bits, enable the transmitter (TEN = 1), and write the outgoing data to the DATA register. Hardware will automatically transfer the data from the DATA register to the transmit FIFO. A FIFO entry is immediately loaded from the FIFO into the shift register any time data is available in the transmit FIFO and the shift register is empty.

The data transmission begins when all of the following conditions are met:

1. The shift register is loaded with data.
2. The transmitter is enabled (TEN = 1).
3. The transmitter is not inhibited (TINH = 0).
4. The module is not halted because of a debug breakpoint if DBGMD is set to 1.
5. The TX output enable is set (TXOEN = 1).

The device may still enter power mode PM8 when the UART is transmitting, and an enabled UART interrupt can wake the device from PM8.

37.9.1. Transmitter FIFO Management

The UART transmit FIFO is implemented as a circular buffer with four entries, allowing data to be pushed as a continuous stream. The DATA register forms a single port into the transmit and receive FIFOs. Writes to the DATA register push data into the transmit FIFO, and reads from the DATA register pop data from the receive FIFO.

The transmitter supports byte, half-word, or word writes to the FIFO. Writes to the FIFO should always be right-justified, so byte-wide writes should always write DATA[7:0], half-word writes should always write DATA[15:0], and word writes should always write DATA[31:0]. The transmitter ignores all other writes to the DATA register, including left-justified byte writes or writes containing more data than will fit in the empty slots in the FIFO. Any illegal FIFO write sets the transmit FIFO error interrupt flag (TFERI) and causes an interrupt, if enabled.

The TDATLN bit controls how the written data is mapped into the FIFO. The transmitter supports data lengths ranging from 5 to 9 bits; the optional start, parity, and stop bits are not stored in the FIFO. When transmitting less than 9 bits of data, the hardware transmits all 5 to 8 bits written to the FIFO by firmware.

When transmitting 9-bit data, the transmitter supports two modes of operation:

1. In normal 9-bit mode (TDATLN = 4), firmware writes 9-bit data to the FIFO with each half-word containing a right-justified 9-bit entry.
2. In fixed 9-bit mode (TDATLN = 5), bus writes to the FIFO are assumed to contain only the least-significant 8 bits of each data entry, and hardware inserts the 9th bit value (set by TBIT) into the FIFO on each FIFO write.

When the data length is 8 bits (or 9 bits when using TDATLN = 5), one, two, or four FIFO entries can be written in a single bus operation by using byte, half-word, or word operations, respectively. For example, writing a whole word (4 bytes) to DATA[31:0] will write four entries in the FIFO with each byte as an entry.

When the data length is 9 bits (using TDATLN = 4), two FIFO entries can be written in a single bus operation by using a word operation, or one entry can be written using a half-word operation. With a half-word write, the hardware will push DATA[8:0] to the FIFO. With a word write, the hardware will push DATA[8:0] to the FIFO, followed by the value in DATA[24:16]. The other bits in the DATA register in this mode are unused and have no effect.

The transfer FIFO count (TCNT) field maintains a count of the number of occupied entries in the transmit FIFO. Firmware may read TCNT to determine the allowable width of writes (byte, half-word, or word) given the available number of empty FIFO entries.

The transmit FIFO threshold (TFTH) field configures the threshold at which the hardware asserts the transmit data request interrupt (TDREQI) and may be configured to 1, 2, or 4 empty FIFO slots.

Firmware can set the transmit FIFO flush (TFIFOFL) bit to flush the contents of the transmit FIFO. A FIFO flush will also clear the contents of the shift register if the transmitter is idle and is not actively transmitting data.

37.9.2. Transmitter Status

The UART module has two status bits which may be monitored to determine the status of the transmitter.

37.9.2.1. Transmitter Busy Flag (TBUSYF)

The transmitter busy flag (TBUSYF) will be asserted when a single byte transmission is in progress. Note that if transmitted multiple bytes with IPDELAY>0, TBUSYF will return to 0 between each transmitted byte. For this reason, TBUSYF is not a reliable indicator of a completed transmission, and the method described below using the TCPTI bit is recommended.

37.9.2.2. Transmitter Complete Interrupt Flag (TCPTI)

The transmit complete interrupt (TCPTI) flag indicates the following:

1. If TCPTTH = 0, this flag indicates a single transmit completed since TCPTI was last cleared.
2. If TCPTTH = 1, this flag indicates a transmit of the last available data in the FIFO completed since TCPTI was last cleared.

To determine when the transmitter has completed transmission of all data in the FIFO, firmware may set TCPTTH=1, write the data into the FIFO, and then wait for the TCPTI bit to be asserted. Note that TCPTI must be manually cleared by firmware.

37.10. Receiving Data

Firmware should first configure all necessary receiver configuration bits before enabling the receiver (REN = 1). Incoming data will be de-glitched and then shifted into the receive shift register. When the data reception is complete, the data will be aligned and padded according to the data length specified in the RDATLN bit field, and then pushed into the circular receive FIFO buffer if an empty entry is available. When both the receive FIFO and the shift register are full, any subsequent received data replaces the shift register contents and causes the hardware to set the receive overrun error interrupt (ROREI) flag.

Firmware may read from the FIFO any time entries are available regardless of the state of the receiver.

The receiver supports one-shot receptions, where the receiver accepts a single byte. Firmware can enable this by setting the ROSEN bit. The hardware automatically clears this bit after the next reception is completed and accepted. If MATMD is set to 1 for MCE mode and a match does not occur, the hardware will not accept the data and will not clear the ROSEN bit.

37.10.1. Receiver FIFO Management

The receive FIFO is implemented as a circular buffer with four entries. The receive and transmit FIFOs are both accessed using the same DATA register; a read from the DATA register pops data from the receive FIFO, and a write to the DATA register pushes data into the transmit FIFO. The receiver supports byte, half-word, or word bus reads from the FIFO. When reading from the FIFO, the least-significant entry is popped from the FIFO first.

The RDATLN field determines how the data is mapped into the FIFO. The receiver supports data lengths ranging from 5 to 9 bits; the optional start, parity, and stop bits are not stored in the FIFO. When receiving less than 9 bits, a firmware read from the FIFO returns all 5 to 8 bits of the receive data.

When receiving 9-bit data, the receiver supports two modes of operation:

1. In normal 9-bit mode (RDATLN = 4), firmware reads 9-bit data from the FIFO, with each half-word containing a right-justified 9-bit entry.
2. In fixed 9-bit mode (RDATLN = 5), only the least-significant 8 bits of received data are stored in the FIFO. The 9th bit is compared against RBIT to generate an interrupt or set error flags according to the MATMD field.

When the data length is 8 bits (or 9 bits when RDATLN = 5), firmware can read one, two, or four FIFO entries in a single bus operation by using byte, half-word, or word operations, respectively.

When the data length is 9 bits (using normal 9-bit mode with RDATLN = 4), each half-word is treated as a right-justified entry. When reading 9-bit data using a half-word bus operation, the hardware places the first entry popped from the FIFO into DATA[8:0]. When reading a whole word, hardware places the first entry popped from the FIFO into DATA[8:0] and the second entry into DATA[24:16]. The rest of the DATA bits are unused in this mode and are set to zero.

The receive FIFO count (RCNT) field maintains a count of the number of occupied entries in the receive FIFO. Firmware may read RCNT to determine the allowable read widths given the available number of full FIFO entries. The receive FIFO threshold (RFTH) field configures the threshold at which hardware asserts a read request interrupt (RDREQI) and may be configured to 1, 2, or 4 occupied entries. Up to four data bytes can be received in PM8 before waking the device.

Firmware can set the receive FIFO flush (RFIFOFL) bit to flush the contents of the receive FIFO. A FIFO flush will also clear the contents of the shift register if the receiver is idle and is not actively receiving data.

37.10.2. Receiver Status

The UART module includes a status bit which may be monitored to determine the status of the receiver.

37.10.2.1. Receiver Busy Flag (RBUSYF)

The receiver busy flag (RBUSYF) will be asserted when a single byte receive is in progress. Note that when receiving multiple bytes, any delay between the bytes may cause RBUSYF to return to 0 between each received byte.

37.11. Additional Communication Support

37.11.1. Multi-Master Mode

The transmitter supports applications in which multiple transmitters are driving the same TX signal. This multi-master mode is suited for applications using LIN, which performs one-wire, half-duplex communications.

Multi-master mode requires the use of idle signal tristates (ITSEN = 1). When this bit is set to 1 and the transmitter is idle, the transmitter automatically tristates the TX pin. The TX pin must have an external pull-up resistor to ensure the pin remains in an idle state when tristated.

37.11.2. Multi-Processor Communications

In a multi-processor application (MCE mode), a master first transmits an address byte to select the target. In an address byte, the 9th bit is always set to a logic 1; in a data byte, the 9th bit is always set to logic 0. The UART module supports multi-processor communication through the use of match operations. Although the match operations are most commonly used with the 9th data bit as an address indicator, the match operations are available regardless of the actual configured data length.

Match operations are enabled or disabled in the receiver by configuring the match mode (MATMD) bit field.

When MATMD is set to 1, the hardware operates in MCE mode and only accepts and stores the incoming receive data if the last data bit matches the value of RBIT. Otherwise, the hardware ignores incoming data, no interrupts are generated, and no FIFO activity occurs. This mode can be used to only accept data when the last data bit marks the frame as containing an address value. Firmware would then compare this address against an assigned value and reconfigure the receiver as needed if the values match. The MATMD setting is automatically switched to Frame mode after a match occurs in the MCE setting.

In Frame mode (MATMD = 2), all incoming data is accepted and stored, but a receive frame error (RFRMERI) asserts if the last data bit matches the value of RBIT. This operation can be used to assert an error when an unexpected address frame arrives.

In Store mode (MATMD = 3), the hardware accepts all incoming data and stores the last data bit in the RBIT field. The last data bit may also be stored in the FIFO depending on the value of RDATLN. For example, if RDATLN is set to 9 bits with the 9th bit stored in the FIFO (RDATLN = 4), and MATMD is set to store, the 9th data bit will be stored in both RBIT and the FIFO.

37.11.3. IrDA

The UART module transmitter and receiver support simple, asynchronous communications with IrDA devices. Note that IrDA support is not available when the UART is used with the RTC0TCLK.

37.11.3.1. IrDA Modulation

Firmware can place the transmitter in IrDA mode by setting the TIRDAEN bit. In IrDA mode, the transmitter performs a RZ (return-to-zero) modulation on the TX signal.

By default, transmitting a 0 causes a pulse to be generated low for a fraction of the bit time and transmitting a 1 leaves TX high without generating a low pulse. The transmit IrDA pulse width (TIRDAPW) bit field configures the pulse width as either 1/16, 1/8, 3/16, or 1/4 of the bit time.

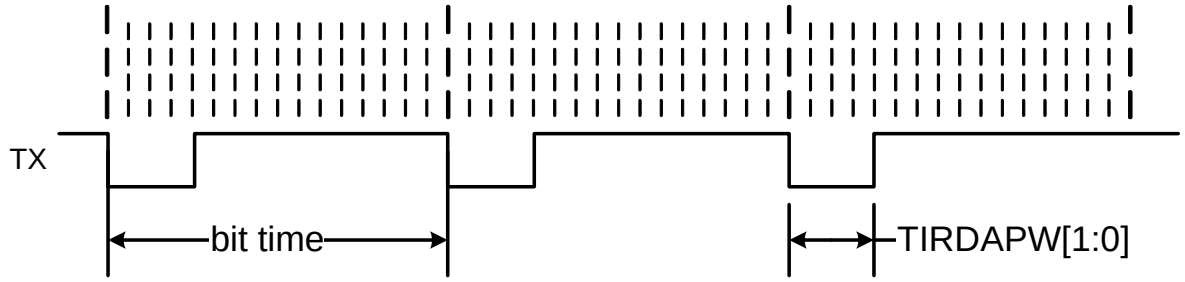


Figure 37.4. UART Transmit IrDA Pulse Width Timing

As shown in Figure 37.4, the default TX idles high and generates a low pulse when a 0 is transmitted. The polarity of TX may be inverted by setting the transmitter invert enable (TINVEN) bit, causing TX to idle low and generate a high pulse when sending a 0.

37.11.3.2. IrDA Demodulation

Setting the receiver IrDA enable (RIRDAEN) to 1 places the receiver in IrDA mode. In this mode, the receiver performs a simple RZ-to-NRZ (return-to-zero to non-return-to-zero) demodulation on the RX signal.

37.11.4. LIN Support

The UART transmitter and receiver can be used with LIN (Local Interconnect Network) systems. Note that LIN support is not available when the UART is used with the RTC0TCLK.

When using LIN, firmware should configure the transmitter to use 8 data bits, 1 start bit, 1 stop bit, and no parity bits. The receiver auto-baud detection also supports the LIN SYNC byte (0x55).

37.11.5. Half Duplex Mode

The UART module supports applications in which half-duplex communications occur across a shared TX/RX signal. In half-duplex mode, the hardware automatically inhibits the receiver during transmit operations and the transmitter during receive operations.

The duplex mode bit (DUPLEXMD) enables half-duplex mode for the module. Firmware should also enable idle tristates (ITSEN = 1) to force the transmitter to tristate the TX signal when not transmitting. The RX and TX pins must be shorted externally in half-duplex mode.

37.11.6. Loop Back Support

The UART module supports several internal loop back options for testing purposes. The loop back modes are configured in the loop back mode (LBMD) field. Table 37.3 describes the different loop back options.

Table 37.3. Internal Loop Back Modes

LBMD value	Loop Back Mode
0	Disabled
1	Receive Loop Back
2	Transmit Loop Back
3	Full Loop Back

When loop back mode is disabled, the module operates normally.

In receive loop back mode, the receiver input path is disconnected from the RX signal and internally connected to the transmitter. Any data transmitted in this mode will be sent out on the TX signal and also received by the device. The physical RX pin tristates in this mode. TX is connected to the corresponding external pin, if the signal is enabled.

With transmit loop back, the transmitter output path is disconnected from the TX signal, and the RX input signal is internally looped back to the TX pin. Data received on the RX signal will be received by the device and also sent directly back out on the TX pin. RX is connected to the corresponding external pin.

In full loop back mode, the transmitter output is internally routed back to the receiver input. Neither the transmitter nor receiver are connected to external device pins. The RX device pin is similarly looped back to the TX pin. Any data transmitted on TX will be sent directly back in on RX.

Figure 37.5 illustrates the internal connections for each loop back mode.

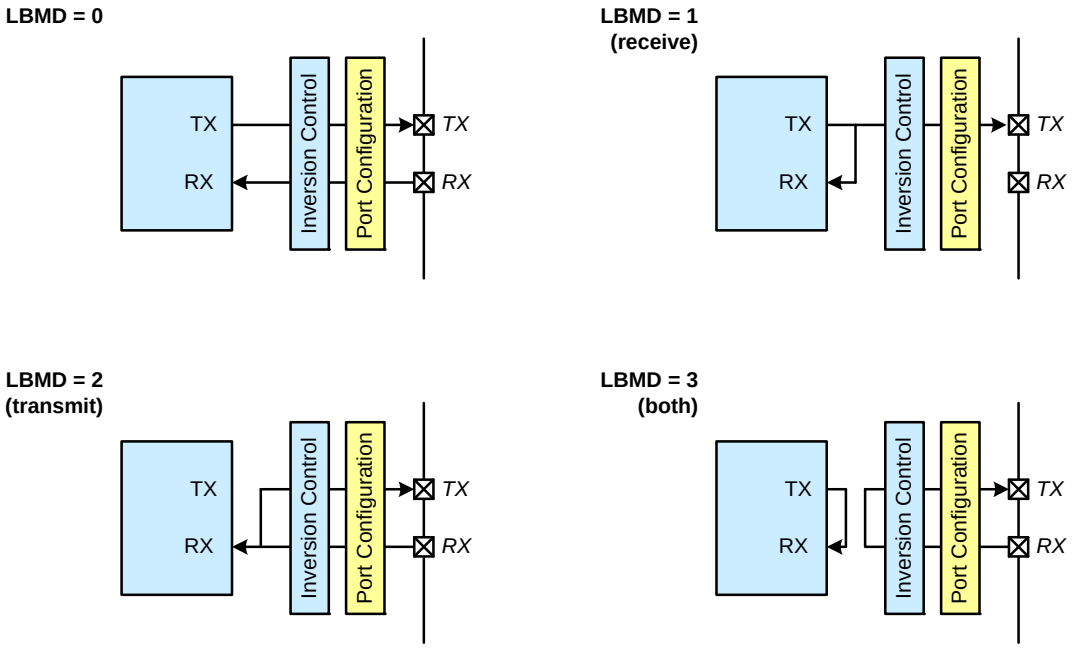


Figure 37.5. Internal Loop Back Connections

37.12. UART0 Registers

This section contains the detailed register descriptions for UART0 registers.

Register 37.1. UART0_CONFIG: Module Configuration

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	TINVEN	TIRDAEN	Reserved		TDATLN			Reserved	TPARMD		TSTPMD		TSTPEN	TPAREN	TSTRTEN
Type	RW	RW	RW	RW	R	RW			R	RW		RW		RW	RW	RW
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved	RINVEN	RIRDAEN	Reserved		RDATLN			Reserved	RPARMD		RSTPMD		RSTPEN	RPAREN	RSTRTEN
Type	RW	RW	RW	RW	R	RW			R	RW		RW		RW	RW	RW
Reset	0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1

Register ALL Access Address

UART0_CONFIG = 0x4000_1000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 37.4. UART0_CONFIG Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	TINVEN	**Transmitter Invert Enable.** 0: Do not invert the TX pin signals (the TX idle state is high). 1: Invert the TX pin signals (the TX idle state is low).
29	TIRDAEN	**Transmitter IrDA Enable.** 0: Disable IrDA transmit mode. 1: Enable IrDA transmit mode.
28:27	Reserved	Must write reset value.

Table 37.4. UART0_CONFIG Register Bit Descriptions

Bit	Name	Function
26:24	TDATLN	**Transmitter Data Length.** Select the number of data bits sent during transmission. 000: 5 bits. 001: 6 bits. 010: 7 bits. 011: 8 bits. 100: 9 bits. The 9th bit is taken from the FIFO data (normal mode). 101: 9 bits. The 9th bit is set by the value of TBIT (fixed mode). 110-111: Reserved.
23	Reserved	Must write reset value.
22:21	TPARMD	**Transmitter Parity Mode.** This field selects the type of parity sent during transmissions. 00: Odd Parity. 01: Even Parity. 10: Set (Parity = 1). 11: Clear (Parity = 0).
20:19	TSTPMD	**Transmitter Stop Mode.** This bit selects the transmitted stop bit length. 00: 0.5 stop bit. 01: 1 stop bit. 10: 1.5 stop bits. 11: 2 stop bits.
18	TSTPEN	**Transmitter Stop Enable.** 0: Do not send stop bits during transmissions. 1: Send stop bits during transmissions.
17	TPAREN	**Transmitter Parity Enable.** 0: Do not send a parity bit during transmissions. 1: Send a parity bit during transmissions.
16	TSTRTEN	**Transmitter Start Enable.** 0: Do not generate a start bit during transmissions. 1: Generate a start bit during transmissions.
15	Reserved	Must write reset value.
14	RINVEN	**Receiver Invert Enable.** 0: Do not invert the RX pin signals (the RX idle state is high). 1: Invert the RX pin signals (the RX idle state is low).
13	RIRDAEN	**Receiver IrDA Enable.** 0: The receiver does not operate in IrDA mode. 1: The receiver operates in IrDA mode.
12:11	Reserved	Must write reset value.

Table 37.4. UART0_CONFIG Register Bit Descriptions

Bit	Name	Function
10:8	RDATLN	**Receiver Data Length.** Select the expected length of received data. 000: 5 bits. 001: 6 bits. 010: 7 bits. 011: 8 bits. 100: 9 bits. The 9th bit is stored in the FIFO (normal mode). 101: 9 bits. The 9th bit is not stored in the FIFO (fixed mode). This mode is used when the 9th bit is only used for match operations (see MATMD). 110-111: Reserved.
7	Reserved	Must write reset value.
6:5	RPARMD	**Receiver Parity Mode.** This field selects the type of parity expected during reception. 00: Odd Parity. 01: Even Parity. 10: Set (Parity = 1). 11: Clear (Parity = 0).
4:3	RSTPMD	**Receiver Stop Mode.** Select the number of stop bits expected during reception. 00: 0.5 stop bit. 01: 1 stop bit. 10: 1.5 stop bits. 11: 2 stop bits.
2	RSTPEN	**Receiver Stop Enable.** 0: Do not expect stop bits during receptions. 1: Expect stop bits during receptions.
1	RPAREN	**Receiver Parity Enable.** 0: Do not expect a parity bit during receptions. 1: Expect a parity bit during receptions.
0	RSTRTEN	**Receiver Start Enable.** 0: Do not expect a start bit during receptions. 1: Expect a start bit during receptions.

Register 37.2. UART0_MODE: Module Mode Select

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved	ITSEN	Reserved		DUPLEXMD	Reserved							LBMD		Reserved	DBGMD
Type	RW	RW	RW		RW	R			RW			R	RW		R	RW
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved		TXCLKSW	RXCLKSW	CLKBUSY	FORCECLK	RTCBDMD	RTCCKMD	Reserved							
Type	R		RW	RW	R	RW	RW	RW	R							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

UART0_MODE = 0x4000_1010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 37.5. UART0_MODE Register Bit Descriptions

Bit	Name	Function
31	Reserved	Must write reset value.
30	ITSEN	**Idle TX Tristate Enable.** 0: The TX pin is always an output in this mode, even when idle. 1: The TX pin is tristated when idle.
29:28	Reserved	Must write reset value.
27	DUPLEXMD	**Duplex Mode.** If this bit is set to 1, the ITSEN bit must also be set to 1 to tristate the TX signal during receive operations. 0: Full-duplex mode. The transmitter and receiver can operate simultaneously. 1: Half-duplex mode. The transmitter automatically inhibits when the receiver is active and the receiver automatically inhibits when the transmitter is active.
26:20	Reserved	Must write reset value.

Table 37.5. UART0_MODE Register Bit Descriptions

Bit	Name	Function
19:18	LBMD	**Loop Back Mode.** Select from different internal loop-back options for diagnostic and debug purposes. 00: Loop back is disabled and the TX and RX signals are connected to the corresponding external pins. 01: Receive loop back. The receiver input path is disconnected from the RX pin and internally connected to the transmitter. Data transmitted will be sent out on TX and also received by the device. 10: Transmit loop back. The transmitter output path is disconnected from the TX pin and the RX input pin is internally looped back out to the TX pin. Data received at RX will be received by the device and also sent directly back out on TX. 11: Full loop back. Internally, the transmitter output is routed back to the receiver input. Neither the transmitter nor receiver are connected to external device pins. The device pin RX is looped back to TX in a similar fashion. Data transmitted on TX will be sent directly back in on RX.
17	Reserved	Must write reset value.
16	DBGMD	**UART Debug Mode.** 0: The UART module will continue to operate while the core is halted in debug mode. 1: A debug breakpoint will cause the UART module to halt. Any active transmissions and receptions will complete first.
15:14	Reserved	Must write reset value.
13	TXCLKSW	**Transmit Automatic Clock Switch.** When set, the UART will automatically switch from the RTC0TCLK to the APB clock source when a transmit interrupt is pending. The least significant two bits of the TBAUD bitfield should be compatible with both the RTC0TCLK and APB clock modes if this feature is used. 0: UART will always use the selected clock for transmit operations. 1: UART will automatically switch from RTC0TCLK to the APB clock when a transmit interrupt is pending.
12	RXCLKSW	**Receive Automatic Clock Switch.** When set, the UART will automatically switch from the RTC0TCLK to the APB clock source when a receive interrupt is pending. The least significant two bits of the RBAUD bitfield should be compatible with both the RTC0TCLK and APB clock modes if this feature is used. 0: UART will always use the selected clock for receive operations. 1: UART will automatically switch from RTC0TCLK to the APB clock when a receive interrupt is pending.
11	CLKBUSY	**Clock Switch Busy Status.** This bit indicates that the UART clock is actively switching between two clock sources. A clock switch may occur if automatic clock switching is enabled (TXCLKSW = 1 or RXCLKSW = 1) or if the RTCCKMD bit is manually changed. Software should check this bit before disabling any unused clock sources to the UART peripheral. 0: Clock switch completed. 1: Clock switch in progress.

Table 37.5. UART0_MODE Register Bit Descriptions

Bit	Name	Function
10	FORCECLK	**Force Clock On.** This bit can be set to force the UART clock on at all times. When the UART is a low-power wakeup source, this bit should be cleared to 0. 0: UART clock is only on when necessary. 1: Force the UART clock to always be on.
9	RTCBDMD	**RTC Baud Rate Mode.** 0: The RBAUD and TBAUD controls use the RTCCKMD setting to determine whether to use APB clock mode (RTCCKMD = 0) or the RTC0TCLK mode (RTCCKMD = 1). Use this setting when APB clock != RTC0TCLK. 1: The RBAUD and TBAUD controls use RTC0TCLK mode. Use this setting when APB clock = RTC0TCLK and RTCCKMD = 0 to force the RBAUD and TBAUD controls into RTC0TCLK mode.
8	RTCCKMD	**RTC Clock Mode.** 0: UART clocked from the APB clock. The RBAUD and TBAUD controls will use the APB clock mode to determine the baudrate unless RTCBDMD = 1. 1: UART clocked from RTC0TCLK. The RBAUD and TBAUD controls will use the RTC0TCLK mode to determine the baudrate. Software should only set this bit to one when the UART is idle.
7:0	Reserved	Must write reset value.

Register 37.3. UART0_FLOWCN: Flow Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved		TIRDAPW		Reserved										TX	Reserved
Type	R		RW		R				RW	R	RW	R		RW	RW	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved			TXOEN	Reserved										RX	Reserved
Type	R			RW	R				RW			R			R	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	X	1

Register ALL Access Address

UART0_FLOWCN = 0x4000_1020

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 37.6. UART0_FLOWCN Register Bit Descriptions

Bit	Name	Function
31:30	Reserved	Must write reset value.
29:28	TIRDAPW	**Transmit IrDA Pulse Width.** This field sets the IrDA pulse width as a fraction of the bit period. 00: The IrDA pulse width is 1/16th of a bit period. 01: The IrDA pulse width is 1/8th of a bit period. 10: The IrDA pulse width is 3/16th of a bit period. 11: The IrDA pulse width is 1/4th of a bit period.
27:18	Reserved	Must write reset value.
17	TX	**TX State.** Firmware can write this bit to change the TX state only when the transmitter is disabled (TEN = 0). 0: The TX pin (before optional inversion) is low. 1: The TX pin (before optional inversion) is high.
16:13	Reserved	Must write reset value.
12	TXOEN	**TX Output Enable.** 0: The pin assigned to TX is controlled by the direct port output value. 1: The pin assigned to TX is controlled by the UART.
11:2	Reserved	Must write reset value.

Table 37.6. UART0_FLOWCN Register Bit Descriptions

Bit	Name	Function
1	RX	**RX Pin Status.** 0: RX pin (after optional inversion) is low. 1: RX pin (after optional inversion) is high.
0	Reserved	Must write reset value.

Register 37.4. UART0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TEN	TINH	Reserved	TBIT	TBUSYF	Reserved			TCPTIEN	TDREQIEN	Reserved	TCPTTH	TCPTI	TDREQI	Reserved	
Type	RW	RW	R	RW	R	R			RW	RW	RW	RW	RW	R	RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	REN	RINH	ROSEN	RBIT	RBUSYF	RABDEN	MATMD		Reserved	RDREQIEN	RERIEN	Reserved	RDREQI	ROREI	RPARERI	RFRMERI
Type	RW	RW	RW	RW	R	RW	RW		R	RW	RW	R	R	RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

UART0_CONTROL = 0x4000_1030

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 37.7. UART0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	TEN	**Transmitter Enable.** 0: Disable the transmitter. When cleared, the transmitter immediately aborts any active transmission. Clearing this bit does not automatically flush the transmit FIFO. 1: Enable the transmitter. The transmitter will initiate a transmission when data becomes available in the transmit FIFO.
30	TINH	**Transmit Inhibit.** 0: The transmitter operates normally. 1: Transmissions are inhibited. The transmitter will stall after any current transmission is complete.
29	Reserved	Must write reset value.
28	TBIT	**Last Transmit Bit.** This bit is used to set the 9th bit during each FIFO write when TDATLN is set to 5. The TBIT value is stored in the FIFO during each FIFO write; the TBIT value at the time of transmission is not used. If TDATLN is not set to 5, the written bit from the core interface is used.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 37.7. UART0_CONTROL Register Bit Descriptions

Bit	Name	Function
27	TBUSYF	**Transmitter Busy Flag.** 0: The UART transmitter is idle. 1: The UART transmitter is active and transmitting.
26:24	Reserved	Must write reset value.
23	TCPTIEN	**Transmit Complete Interrupt Enable.** 0: Disable the transmit complete interrupt. 1: Enable the transmit complete interrupt. A transmit interrupt is generated when TCPTI is set to 1.
22	TDREQIEN	**Transmit Data Request Interrupt Enable.** 0: Disable the transmit data request interrupt. 1: Enable the transmit data request interrupt. A transmit interrupt is asserted when TDREQI is set to 1.
21	Reserved	Must write reset value.
20	TCPTTH	**Transmit Complete Threshold.** 0: The TCPTI flag is set after each data transmission. 1: The TCPTI flag is set after transmission of the last available data.
19	TCPTI	**Transmit Complete Interrupt Flag.** This bit is set by hardware if a byte is transmitted (TCCPTH = 0) or if the last available byte is transmitted (TCPTTH = 1). This bit must be cleared by firmware.
18	TDREQI	**Transmit Data Request Interrupt Flag.** 0: The transmitter is not requesting more FIFO data. 1: The transmitter is requesting more FIFO data.
17:16	Reserved	Must write reset value.
15	REN	**Receiver Enable.** This bit enables the UART receiver for multiple transactions. 0: Disable the receiver. The receiver can receive one data transaction only if ROSEN is set. 1: Enable the receiver.
14	RINH	**Receiver Inhibit.** 0: The receiver operates normally. 1: The receiver will complete any ongoing reception, but ignore all traffic after that.
13	ROSEN	**Receiver One-Shot Enable.** 0: Disable one-shot receive mode. 1: Enable one-shot receive mode.
12	RBIT	**Last Receive Bit.** This bit is used according to the match mode (MATMD) setting.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 37.7. UART0_CONTROL Register Bit Descriptions

Bit	Name	Function
11	RBUSYF	**Receiver Busy Flag.** 0: The UART receiver is idle. 1: The UART receiver is receiving data.
10	RABDEN	**Receiver Auto-Baud Enable.** 0: Disable receiver auto-baud. 1: Enable receiver auto-baud.
9:8	MATMD	**Match Mode.** The hardware automatically switches from MCE mode to Frame mode when a MCE match occurs. 00: Disable the match function. 01: (MCE) Data whose last data bit equals RBIT is accepted and stored. 10: (Frame) A framing error is asserted if the last received data bit matches RBIT. 11: (Store) Store the last incoming data bit in RBIT. This mode can be used incon-junction with the RDATLN setting.
7	Reserved	Must write reset value.
6	RDREQIEN	**Receive Data Request Interrupt Enable.** 0: Disable the read data request interrupt. 1: Enable the read data request interrupt. A receive interrupt is generated when RDREQI is set to 1.
5	RERIEN	**Receive Error Interrupt Enable.** 0: Disable the receive error interrupt. 1: Enable the receive error interrupt. A receive interrupt is asserted when ROREI, RFRMERI, or RPARERI is set to 1.
4	Reserved	Must write reset value.
3	RDREQI	**Receive Data Request Interrupt Flag.** 0: Fewer than RFTH FIFO entries are filled with data. 1: At least RFTH FIFO entries are filled with data.
2	ROREI	**Receive Overrun Error Interrupt Flag.** This bit is set to 1 by hardware when a receive overrun error occurs. This bit must be cleared by firmware.
1	RPARERI	**Receive Parity Error Interrupt Flag.** This bit is set to 1 by hardware when the receiver encounters a parity error. This bit must be cleared by firmware.
0	RFRMERI	**Receive Frame Error Interrupt Flag.** Hardware sets this bit to 1 when a receive frame error occurs. There are two sources for this error: when an expected whole stop bit is low, or when a frame match-mode fails. This bit must be cleared by firmware.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Register 37.5. UART0_IPDELAY: Inter-Packet Delay

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved								IPDELAY							
Type	R								RW							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

UART0_IPDELAY = 0x4000_1040

Table 37.8. UART0_IPDELAY Register Bit Descriptions

Bit	Name	Function
31:24	Reserved	Must write reset value.
23:16	IPDELAY	**Inter-Packet Delay.** This field configures the transmitter to delay between transmissions by the specified number of bit times. A value of 0 means no delay is added, and a value of 5 means 5 bit times are added. Bit times are configured in the TBAUD register.
15:0	Reserved	Must write reset value.

Register 37.6. UART0_BAUDRATE: Transmit and Receive Baud Rate

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TBAUD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	RBAUD															
Type	RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address
UART0_BAUDRATE = 0x4000_1050

Table 37.9. UART0_BAUDRATE Register Bit Descriptions

Bit	Name	Function
31:16	TBAUD	**Transmitter Baud Rate Control.** APB Clock Mode: This mode is used when the UART clock = APB clock and the APB clock != 32.768kHz (RTCCKMD = 0 and RTCBDMD = 0), In APB Clock Mode, the transmitter baud rate is determined according to the equation: $$\text{Baud Rate} = \frac{F_{APB}}{2^{CLKDIV} \times N \times (TBAUD + 1)}$$ N = 2 if RIRDAEN = 0, and N = 16 if RIRDAEN = 1. RTC0TCLK Mode: This mode is used whenever the UART clock is sourced from a 32.768kHz clock source. This may occur in either of the following two scenarios: 1) The UART clock = RTC0TCLK because the RTC Clock Mode bit is set (RTCCKMD = 1 and RTCBDMD = 0) or 2) The UART clock = APB clock and the APB clock = RTC0TCLK due to the Clock Control module settings (RTCCKMD = 0). In RTC0TCLK mode, only the two least significant bits of the TBAUD register are used. TBAUD[1:0] selects from the following baud rates (assuming a 32.768 kHz RTC0TCLK frequency): 00: 1200 01: 2400 10: 4800 11: 9600

Table 37.9. UART0_BAUDRATE Register Bit Descriptions

Bit	Name	Function
15:0	RBAUD	**Receiver Baud Rate Control.** APB Clock Mode: This mode is used when the UART clock = APB clock and the APB clock != 32.768kHz (RTCCKMD = 0 and RTCBDMD = 0), In APB Clock Mode, the receiver baud rate is determined according to the equation: $\text{Baud Rate} = \frac{F_{APB}}{2^{CLKDIV} \times N \times (RBAUD + 1)}$ RTC0TCLK Mode: This mode is used whenever the UART clock is sourced from a 32.768kHz clock source. This may occur in either of the following two scenarios: 1) The UART clock = RTC0TCLK because the RTC Clock Mode bit is set (RTCCKMD = 1) or 2) The UART clock = APB clock and the APB clock = RTC0TCLK due to the Clock Control module settings (RTCCKMD = 0 and RTCBDMD = 1). In RTC0TCLK mode, only the two least significant bits of the RBAUD register are used. RBAUD[1:0] selects from the following baud rates (assuming a 32.768 kHz RTC0TCLK frequency): 00: 1200 01: 2400 10: 4800 11: 9600

Register 37.7. UART0_FIFOCN: FIFO Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved					TSRFULLF	TFERI	TFIFOFL	Reserved		TFTH		Reserved	TCNT		
Type	R					R	RW	RW	RW	R	RW		R	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved					RSRFULLF	RFERI	RFIFOFL	Reserved		RFTH		Reserved	RCNT		
Type	R					R	RW	RW	RW	R	RW		R	R		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

UART0_FIFOCN = 0x4000_1060

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 37.10. UART0_FIFOCN Register Bit Descriptions

Bit	Name	Function
31:27	Reserved	Must write reset value.
26	TSRFULLF	**Transmit Shift Register Full Flag.** This bit indicates when the transmitter shift register contains data. The bit is set when hardware loads the data from the FIFO to the transmit shift register and is cleared when the transmitter completely transmits the data.
25	TFERI	**Transmit FIFO Error Interrupt Flag.** This bit is set when an illegal FIFO write is detected, such as a non right-justified write or a write that contains more data than will fit in the empty FIFO entries. This bit must be cleared by firmware. 0: A transmit FIFO error has not occurred since TFERI was last cleared. 1: A transmit FIFO error occurred.
24	TFIFOFL	**Transmit FIFO Flush.** Setting this bit to 1 flushes the transmit FIFO. If data is pending in the transmit shift register but a transmit has not begun, the shift register is also flushed. This bit always reads as 0.
23:22	Reserved	Must write reset value.

Table 37.10. UART0_FIFOCN Register Bit Descriptions

Bit	Name	Function
21:20	TFTH	**Transmit FIFO Threshold.** This field sets the FIFO threshold at which a TDREQI interrupt is asserted. 00: A transmit data request interrupt (TDREQI) is asserted when >= 1 FIFO entry is empty. 01: A transmit data request interrupt (TDREQI) is asserted when >= 2 FIFO entries are empty. 10: A transmit data request interrupt (TDREQI) is asserted when >= 3 FIFO entries are empty. 11: A transmit data request interrupt (TDREQI) is asserted when >= 4 FIFO entries are empty.
19	Reserved	Must write reset value.
18:16	TCNT	**Transmit FIFO Count.** This field indicates the number of entries in the transmit FIFO.
15:11	Reserved	Must write reset value.
10	RSRFULLF	**Receive Shift Register Full Flag.** This bit indicates when the receiver shift register contains data and remains high until the data is moved to the FIFO or is flushed. The flag is set as soon as incoming data is completely received and cleared when the data is transferred to the FIFO.
9	RFERI	**Receive FIFO Error Interrupt Flag.** This bit is set when hardware detects an illegal FIFO read, such as a non right-justified read or a read that demands more data than is available in the FIFO. 0: A receive FIFO error has not occurred since RFERI was last cleared. 1: A receive FIFO error occurred.
8	RFIFOFL	**Receive FIFO Flush.** Setting this bit to 1 flushes the receive FIFO and any completed data in the receive shift register. This bit always reads as 0.
7:6	Reserved	Must write reset value.
5:4	RFTH	**Receive FIFO Threshold.** This is the threshold at which a RDREQI interrupt is asserted. 00: A read data request interrupt (RDREQI) is asserted when >= 1 FIFO entry is full. 01: A read data request interrupt (RDREQI) is asserted when >= 2 FIFO entries are full. 10: A read data request interrupt (RDREQI) is asserted when >= 3 FIFO entries are full. 11: A read data request interrupt (RDREQI) is asserted when >= 4 FIFO entries are full.
3	Reserved	Must write reset value.
2:0	RCNT	**Receive FIFO Count.** This field indicates the number of entries in the receive FIFO.

Register 37.8. UART0_DATA: FIFO Input/Output Data

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA[31:16]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	DATA[15:0]															
Type	RW															
Reset	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Register ALL Access Address

UART0_DATA = 0x4000_1070

Table 37.11. UART0_DATA Register Bit Descriptions

Bit	Name	Function
31:0	DATA	**FIFO Data.** The 32-bit DATA register is a single port into the transmit and receive FIFOs. Reads and writes should always be right-justified. Byte-wide reads and writes should always access DATA[7:0], half-word reads and writes should always access DATA[15:0], and word reads and writes should always access DATA[31:0]. Writes to the DATA register push data into the transmit FIFO. Reads from the DATA register pop data from the receive FIFO. Multiple FIFO entries can be pushed or popped in a single write or read. For example, if the transmit bit-length is less than 9 bits, writing a whole word (4 bytes) to the DATA field will write four entries in the FIFO where each byte is a single entry. However, if the FIFO doesn't have room for 4 entries, the write is completely ignored and the TFERI error flag is set. Similarly, a half-word write will push 2 entries to the FIFO if the data length is < 9. If the data length is 9, each half-word is a single entry. When writing or reading multiple bytes from the FIFO, the least significant byte is written to or read from the FIFO first.

Notes:

1. Reads of this register modify the state of hardware. Debug logic should take care when reading this register.

Register 37.9. UART0_CLKDIV: Clock Divider

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved														CLKDIV	
Type	R														RW	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Register ALL Access Address

UART0_CLKDIV = 0x4000_1080

Table 37.12. UART0_CLKDIV Register Bit Descriptions

Bit	Name	Function
31:2	Reserved	Must write reset value.
1:0	CLKDIV	**Clock Divider.** Controls the pre-divider to the UART clock. This field should always be set to divide by 1 when running from the RTC0TCLK. 00: Divide by 1. 01: Divide by 2. 10: Divide by 4. 11: Reserved.

37.13. UART0 Register Memory Map

Table 37.13. UART0 Memory Map

UART0_IPDELAY	UART0_CONTROL	UART0_FLOWCN	UART0_MODE	UART0_CONFIG	Register Name
0x4000_1040	0x4000_1030	0x4000_1020	0x4000_1010	0x4000_1000	ALL Address
ALL	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR	ALL | SET | CLR	Access Methods
Reserved	TEN	Reserved	Reserved	Reserved	Bit 31
	TINH		ITSEN	TINVEN	Bit 30
	Reserved	TIRDAPW	Reserved	TIRDAEN	Bit 29
	TBIT			Reserved	Bit 28
	TBUSYF	Reserved	DUPLEXMD		Bit 27
	Reserved		Reserved	TDATLN	Bit 26
					Bit 25
					Bit 24
IPDELAY	TCPTIEN			Reserved	Bit 23
	TDREQIEN			TPARMD	Bit 22
	Reserved				Bit 21
	TCPTTH		LBMD	TSTPMD	Bit 20
	TCPTI				Bit 19
	TDREQI			TSTPEN	Bit 18
	Reserved	TX	Reserved	TPAREN	Bit 17
			DBGMD	TSTRTEN	Bit 16
	REN	Reserved	Reserved	Reserved	Bit 15
	RINH			RINVEN	Bit 14
	ROSEN		TXCLKSW	RIRDAEN	Bit 13
	RBIT	TXOEN	RXCLKSW	Reserved	Bit 12
	RBUSYF	Reserved	CLKBUSY		Bit 11
	RABDEN		FORCECLK	RDATLN	Bit 10
	MATMD		RTCBDMD		Bit 9
			RTCCKMD		Bit 8
Reserved	Reserved		Reserved	Reserved	Bit 7
	RDREQIEN			RPARMD	Bit 6
	RERIEN				Bit 5
	Reserved			RSTPMD	Bit 4
	RDREQI				Bit 3
	RORFI			RSTPEN	Bit 2
	RPARERI	RX		RPAREN	Bit 1
	RFRMERI	Reserved		RSTRTEN	Bit 0

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 37.13. UART0 Memory Map

Register Name	UART0_CLKDIV	UART0_DATA	UART0_FIFOCN	UART0_BAUDRATE	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
ALL Address	0x4000_1080	0x4000_1070	0x4000_1060	0x4000_1050																														
Access Methods	ALL	ALL	ALL | SET | CLR	ALL																														
	Reserved	DATA	Reserved	TBAUD																														
	Reserved	DATA	TSRFULLF	TBAUD																														
	Reserved	DATA	TFERI	TBAUD																														
	Reserved	DATA	TFIFOFL	TBAUD																														
	Reserved	DATA	Reserved	TBAUD																														
	Reserved	DATA	TFTH	TBAUD																														
	Reserved	DATA	Reserved	TBAUD																														
	Reserved	DATA	TCNT	TBAUD																														
	Reserved	DATA	Reserved	RBAUD																														
	Reserved	DATA	RSRFULLF	RBAUD																														
	Reserved	DATA	RFERI	RBAUD																														
	Reserved	DATA	RFIFOFL	RBAUD																														
	Reserved	DATA	Reserved	RBAUD																														
	Reserved	DATA	RFTH	RBAUD																														
	Reserved	DATA	Reserved	RBAUD																														
	CLKDIV	DATA	RCNT	RBAUD																														

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

38. Voltage Supply Monitor (VMON0)

This section describes the Voltage Supply Monitor (VMON) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the VMON block, which is used by all device families covered in this document.

38.1. Voltage Supply Monitor Features

The Voltage Supply Monitor module includes the following features:

- Main supply “VBAT Low” (VBAT below the early warning threshold) notification.
- Holds the device in reset if the main VBAT supply drops below the VBAT Reset threshold.

The Voltage Supply Monitor allows devices to function in known, safe operating conditions without the need for external hardware.

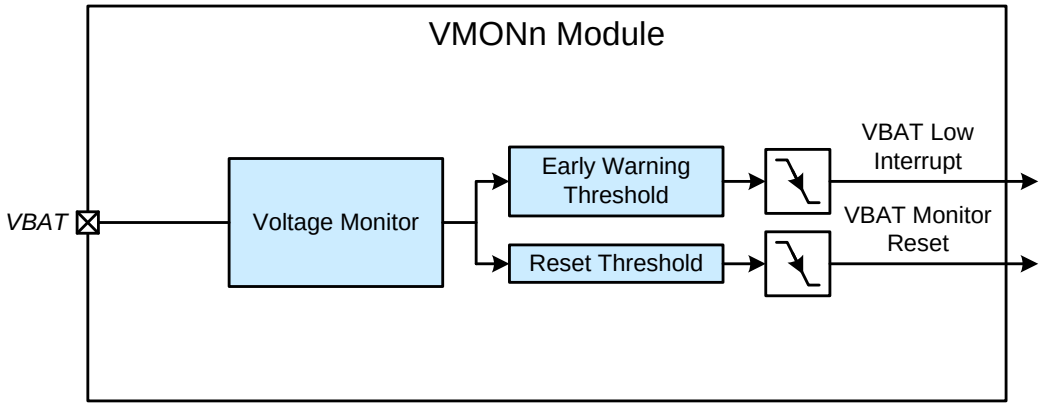


Figure 38.1. Voltage Supply Monitor Block Diagram

38.2. VBAT Supply Monitoring

The VBAT supply monitor senses the voltage on the device VBAT supply and can generate an interrupt or reset if the supply drops below the corresponding thresholds. This monitor is active and enabled as a reset source after initial power-on to protect the device until VBAT is an adequate and stable voltage.

When enabled and selected as a reset source, any power down transition or power irregularity that causes VBAT to drop below the reset threshold will drive the RESET pin low and hold the core in a reset state. When VBAT returns to a level above the reset threshold, the monitor will release the core from the reset state. The reset status can then be read using the device reset sources module. The VBAT monitor reset threshold status can be read using the VBATRSTF flag. The power-on reset delay (t_{POR}) is not incurred after a supply monitor reset.

The contents of standard RAM is invalid after a VBAT monitor reset. Any data stored in retention RAM will be valid unless the supply drops enough for the device to detect a power-on reset. If the device is operating from the low-power charge pump in power mode PM8, the charge pump monitor can also indicate whether RAM may have been corrupted. Firmware should check and interpret the flags in the device reset sources module before relying on retention RAM, to ensure that a power-on reset did not occur.

The enable state of the VBAT supply monitor and its selection as a reset source is not altered by device resets. For example, if the VBAT supply monitor is de-selected as a reset source and disabled by software, and then firmware performs a software reset, the VBAT supply monitor will remain disabled and de-selected after the reset.

To protect the integrity of Flash contents, the VBAT supply monitor must be enabled and selected as a reset source if software contains routines that erase or write Flash memory. If the VBAT supply monitor is not enabled, any erase or write performed on Flash memory will be ignored.

The VBAT monitor also includes a VBAT-is-low interrupt. Hardware clears the VBATLI interrupt flag when VBAT drops below the early warning threshold. This 1-to-0 transition on VBATLI can generate an interrupt when the VBAT low interrupt enable (VBATLIEN) bit is set to 1. The early warning interrupt can be used to save any data in the retention RAM, if supported on the device, and otherwise prepare the system for power down.

The high threshold enable (VBATHITHEN) bit can increase the VBAT monitor reset and early warning thresholds by approximately 300 mV, if appropriate for the system. This setting is recommended when operating at faster AHB clock speeds. The device data sheet contains more information.

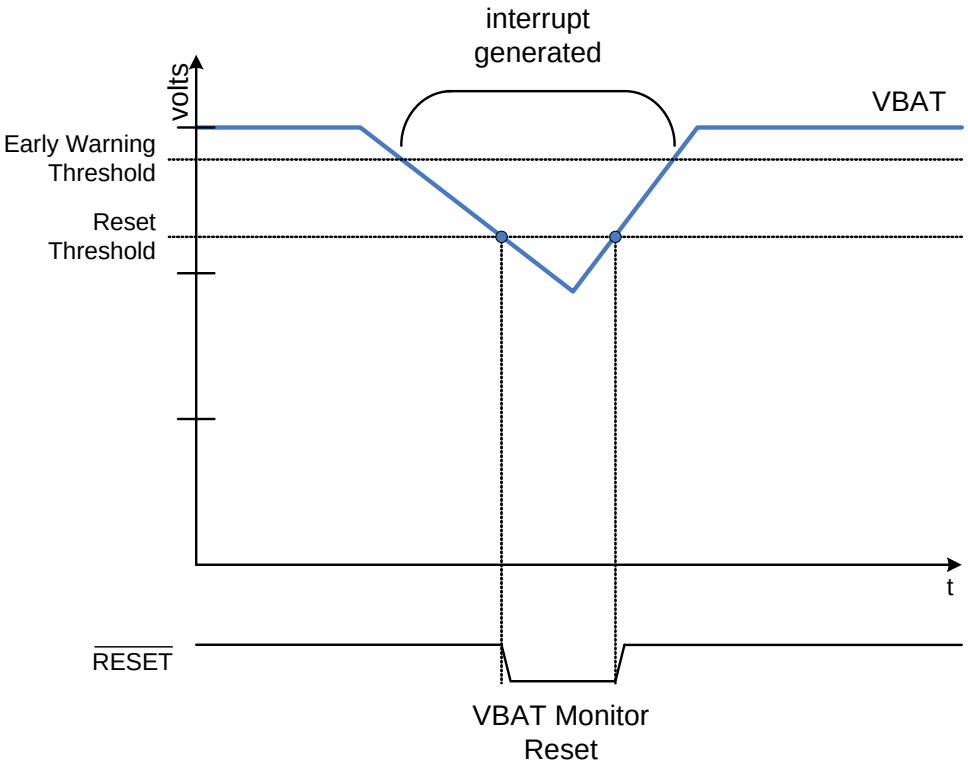


Figure 38.2. VBAT Supply Monitor Thresholds

38.2.1. Enabling the VBAT Monitor

The VBAT supply monitor must be enabled before selecting it as a reset source. Selecting the VBAT supply monitor as a reset source before it has stabilized may generate a system reset. In systems where this reset would be undesirable, a delay should be introduced between enabling the VBAT supply monitor and selecting it as a reset source. No delay should be introduced in systems where software contains routines that erase or write Flash memory. The procedure for enabling the VBAT supply monitor and selecting it as a reset source is:

1. Enable the VBAT supply monitor (VMONEN = 1).
2. Wait for the VBAT supply monitor to stabilize (optional).
3. Select the VBAT monitor as a reset source in the device reset sources module.

38.3. VMON0 Registers

This section contains the detailed register descriptions for VMON0 registers.

Register 38.1. VMON0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VMONEN	Reserved														
Type	RW	R														
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved									VBATLIEN	Reserved	VBATHITHEN	VBATLI	VBATRSTF	Reserved	
Type	R								RW	RW	R	RW	R	R	R	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	X	X	X	0

Register ALL Access Address

VMON0_CONTROL = 0x4002_F000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 38.1. VMON0_CONTROL Register Bit Descriptions

Bit	Name	Function
31	VMONEN	**VBAT Supply Monitor Enable.** This bit enables the voltage supply monitor circuitry to monitor the VBAT supply. 0: Disable the VBAT supply monitor. 1: Enable the VBAT supply monitor.
30:7	Reserved	Must write reset value.
6	VBATLIEN	**VBAT Low Interrupt Enable.** Enables the VBAT low (early warning) interrupt. 0: Disable the VBAT low interrupt. 1: Enable the VBAT low interrupt.
5	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

Table 38.1. VMON0_CONTROL Register Bit Descriptions

Bit	Name	Function
4	VBATHITHEN	**VBAT High Threshold Enable.** Setting this bit to 1 will raise both the VBAT low (early warning) and VBAT reset thresholds by approximately 300 mV. 0: Use the standard VBAT thresholds. 1: Use the high VBAT thresholds.
3	VBATLI	**VBAT Low Interrupt Flag.** This bit indicates whether the VBAT supply voltage is above or below the early warning threshold. Falling below this threshold will generate an interrupt, if the VBAT low interrupt is enabled. 0: The VBAT voltage is below the early warning threshold. 1: The VBAT voltage is above the early warning threshold.
2	VBATRSTF	**VBAT Reset Threshold Status Flag.** This bit indicates whether the VBAT supply voltage is above or below the reset threshold. If the VBAT supply monitor is enabled as a reset source and VBAT falls below the VBAT reset threshold, the device will be reset. 0: The VBAT voltage is below the VBAT reset threshold. 1: The VBAT voltage is above the VBAT reset threshold.
1:0	Reserved	Must write reset value.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.

38.4. VMON0 Register Memory Map

Table 38.2. VMON0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VMON0_CONTROL	0x4002_F000	ALL | SET | CLR	VMONEN	Reserved																								VBATLIEN	Reserved	VBATHITHEN	VBATLI	VBATRSTF	Reserved	

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

39. Voltage Reference and Temperature Sensor (VREF0)

This section describes the Voltage Reference and Temperature Sensor (VREF) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “B” of the VREF block, which is used by all device families covered in this document.

39.1. Voltage Reference Features

The Voltage Reference module includes the following features:

- Two programmable settings: 1.2 and 2.4 V.
- The voltage reference can be used internally while driven on the VREF pin.
- Temperature sensor provides a voltage output that can be measured by an ADC module.

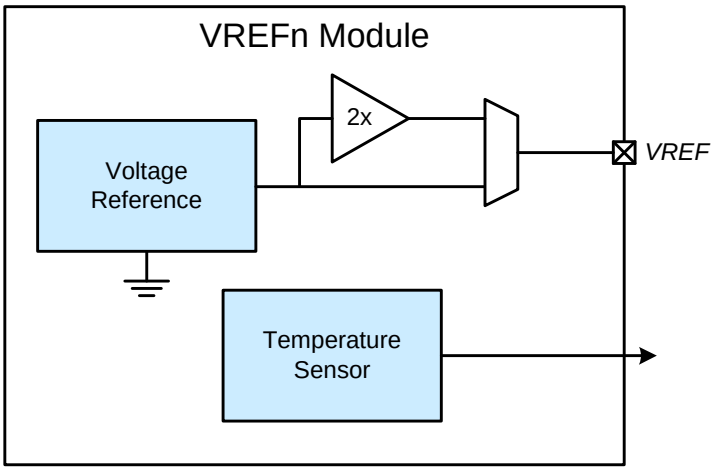


Figure 39.1. Voltage Reference Block Diagram

39.2. Functional Description

The Voltage Reference module contains both a precision bandgap voltage reference generator and a temperature sensor. The reference voltage output is available on the VREF pin. It can be used internally by analog peripherals such as ADCs, and it can also serve as a reference for external circuitry. There are two nominal output levels: 1.2 and 2.4 V, which allows the VREF to be used with different supply voltages.

39.2.1. Voltage Reference Operation

The internal voltage reference can be enabled using the VREFEN bin in the CONTROL register. When enabled, the internal voltage reference node is available to any analog peripherals which have a VREF input. When disabled, the reference supply is completely shut down.

The VREF2X bit in the CONTROL register can be used to double the output voltage of the reference from 1.2 to 2.4 V (nominal). The lower setting is ideal for applications where the supply voltage may be at the lower end of the valid supply range.

The reference voltage output must be sent to the VREF pin on the device when the reference is enabled. When sending the reference voltage to the VREF pin, the VREF pin should be configured for analog mode, and a minimum bypass capacitance of 0.1 µF should be used between VREF and VSS.

39.2.2. Temperature Sensor Operation

The temperature sensor is also controlled from within the VREF module. The temperature sensor is enabled using the TEMPEN bit in the CONTROL register. When enabled, the temperature sensor output is available to any analog peripherals which have a temperature sensor input.

The temperature sensor output is linear with a positive slope; when temperature increases, the output voltage of the sensor increases. Refer to the electrical characteristics section of the specific device data sheet for temperature sensor parameters. Figure 39.2 illustrates the temperature sensor transfer function.

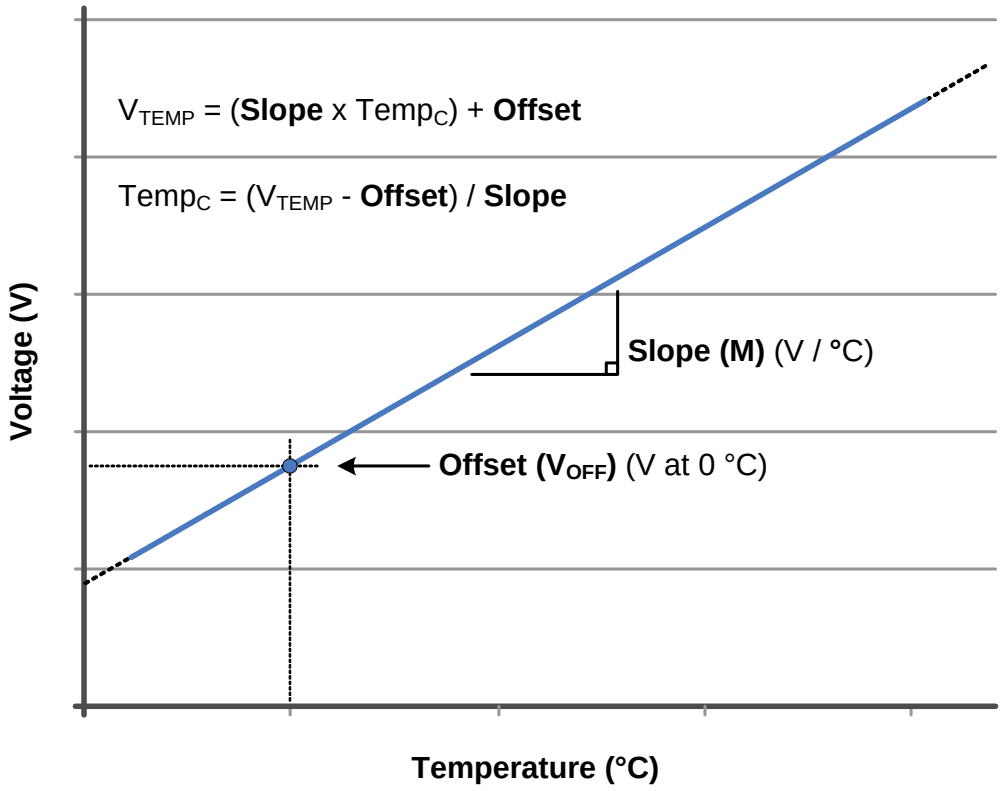


Figure 39.2. Temperature Sensor Transfer Function

39.3. VREF0 and Temperature Sensor Registers

This section contains the detailed register descriptions for VREF0 registers.

Register 39.1. VREF0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved													VREFOUTEN	TEMPEN	VREF2X
Type	R													RW	RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

VREF0_CONTROL = 0x4003_9010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 39.1. VREF0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:3	Reserved	Must write reset value.
2	VREFOUTEN	**VREF Output Enable.** 0: Internal VREF is not driven on the VREF pin. 1: Internal VREF is driven out to the VREF pin.
1	TEMPEN	**Temperature Sensor Enable.** 0: Disable the temperature sensor. 1: Enable the temperature sensor.
0	VREF2X	**Voltage Reference Doubler.** Enables a 2x buffer on the reference to double the output voltage. 0: VREF output is nominally 1.2 V 1: VREF output is nominally 2.4 V

39.4. VREF0 Register Memory Map

Table 39.2. VREF0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
VREF0_CONTROL	0x4003_9010	ALL | SET | CLR	Reserved																													VREFOUTEN	TEMPEN	VREF2X

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

40. Watchdog Timer (WDTIMER0)

This section describes the Watchdog Timer (WDTIMER) module, and is applicable to all products in the following device families, unless otherwise stated:

- SiM3L1xx

This section describes version “A” of the WDTIMER block, which is used by all device families covered in this document.

40.1. Watchdog Timer Features

The Watchdog Timer module includes the following features:

- Programmable timeout interval.
- Optional interrupt to warn when the watchdog timer is nearing the reset trip value.
- Lock-out feature to prevent any modification until a system reset.

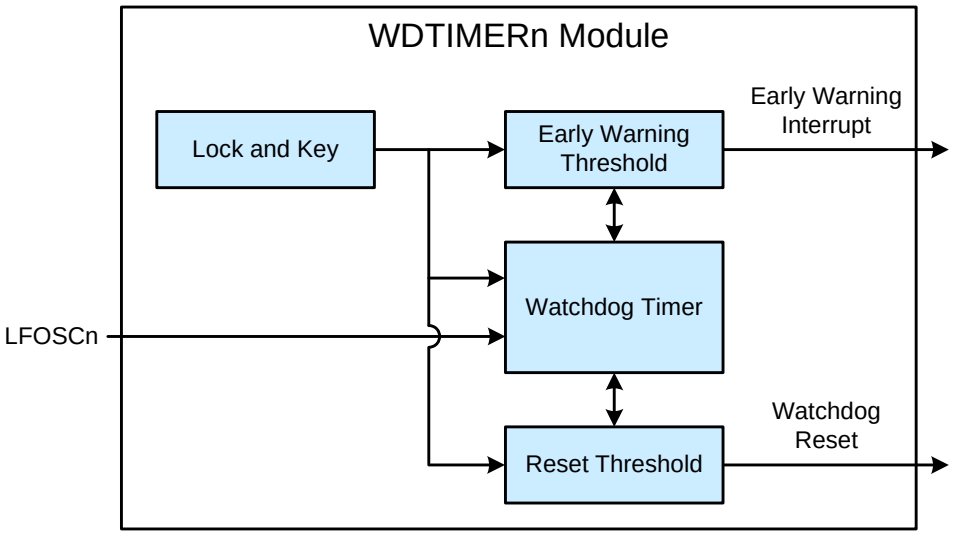


Figure 40.1. Watchdog Timer Block Diagram

40.2. Overview

The WDTIMER module includes a 16-bit timer, a programmable early warning interrupt, and a programmable reset period. The timer registers are protected from inadvertent access by an independent lock and key interface.

The watchdog timer runs from a low frequency oscillator (LFOSC0). Writes to the COMPARE register takes several clock cycles of the associated low frequency oscillator to complete.

40.3. Lock and Key Interface

The lock and key interface locks all WDTIMER register writes to protect firmware from inadvertently modifying any of the timer settings. The WDTKEY register contains the 8-bit KEY field. Firmware must write the attention value followed by one of the valid command values to this key field to interact with the module registers. The KEYSTS bit reports whether firmware must write the attention key or if the module is waiting for a command. Table 40.1 lists the valid key values.

Table 40.1. Valid Key Values

Command	KEY Value
Attention (ATTN)	0xA5
Write (WRITE)	0xF1
Reset (RESET)	0xCC
Disable (DISABLE)	0xDD
Start (START)	0xEE
Lock (LOCK)	0xFF

40.3.1. Write

To issue a write command, firmware must write the attention key followed by a write command to the KEY field. This command allows one write access to one of the watchdog timer registers. Once firmware completes the write, the interface relocks.

The PRIVSTS flag indicates whether the watchdog timer registers are unlocked for a write operation. The WDTIMER registers can be read at any time, regardless of the write status.

40.3.2. Reset

The reset command resets the timer back to zero. This action must be performed periodically to ensure the timer does not cause a system reset. The early warning interrupt can be used as a means to debug the system and determine the cause of unknown WDT resets.

Firmware must first write the attention key followed by the reset command to reset the timer.

40.3.3. Disable

The disable command stops the watchdog timer until the next system reset or until the module receives a Start command.

Firmware must write the attention key followed by the disable command to disable the timer.

40.3.4. Start

To issue a start command, firmware must write the attention key followed by a start command. The start command re-starts the watchdog timer if it has been stopped by a disable command.

40.3.5. Lock

The lock command prevents any disable commands from taking effect until the next system reset. This command ensures that firmware will not accidentally disable the timer. Firmware can still reset the timer and update the thresholds while the timer is locked.

Firmware can issue a lock command by writing the attention key followed by the lock command to the KEY field.

Note that the watchdog timer can still be disabled as a reset source by firmware, even if the timer is locked. To prevent this disable function, firmware should lock the RSTSRC0 registers by setting the CLKRSTL bit in the LOCK0_PERIPHLOCK0 register to 1. The watchdog timer can also be halted if it loses its clock source. To prevent this from happening, the LFOSC registers can be locked in the same manner, using the LPOSCL bit.

40.4. Setting the Early Warning and Reset Thresholds

The early warning interrupt (EWTH) and reset (RTH) thresholds contain the values that can cause an interrupt or system reset. These 16-bit values are directly compared to the 16-bit timer. The early warning threshold can be used to remind firmware to periodically reset the timer.

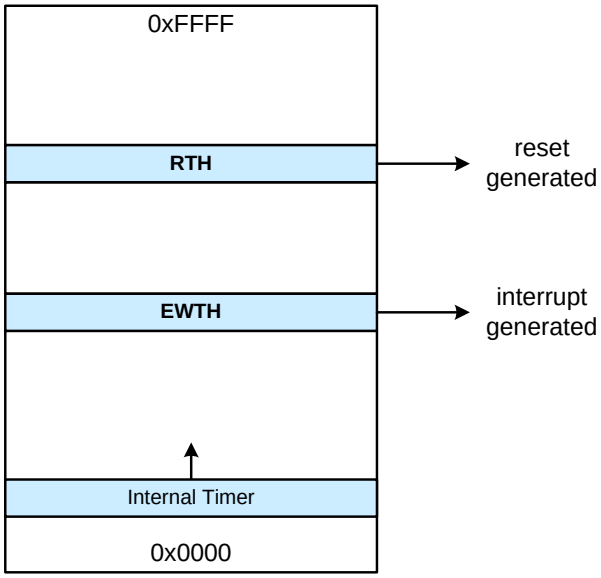


Figure 40.2. Early Warning and Reset Thresholds

Firmware can update these thresholds using the following procedure:

1. Write the attention key (0xA5) to the KEY field.
2. Write the write command (0xF1) to the KEY field.
3. Write the new EWTH and RTH values to the THRESHOLD register in one write operation.
4. (Optional) Poll on the UPDSTS flag to determine when the register update completes.

To write the threshold fields separately, firmware should use the following procedure:

1. Write the attention key (0xA5) to the KEY field.
2. Write the write command (0xF1) to the KEY field.
3. Write the new EWTH value to the THRESHOLD register using a half word access.
4. Poll on the UPDSTS flag to determine when the register update completes.
5. Write the attention key (0xA5) to the KEY field.
6. Write the write command (0xF1) to the KEY field.
7. Write the new RTH value to the THRESHOLD register using a half word access.

40.5. Interrupts and Flags

Hardware sets the early warning interrupt (EWI) flag when the timer reaches the value in the EWTH field. This flag can also cause an interrupt, if enabled (EWIEN = 1).

The reset threshold status (RTHF) flag indicates whether the timer has passed the RTH field value. A system reset can also occur when the timer passes this threshold, if enabled in the device reset sources module.

40.6. Debug Mode

Firmware can set the DBGMD bit to force the WDTIMER module to halt on a debug breakpoint. Clearing the DBGMD bit forces the WDTIMER module to continue operating while the core halts in debug mode.

40.7. WDTIMER0 Registers

This section contains the detailed register descriptions for WDTIMER0 registers.

Register 40.1. WDTIMER0_CONTROL: Module Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved														DBGMD	EWIEN
Type	R														RW	RW
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Register ALL Access Address

WDTIMER0_CONTROL = 0x4003_0000

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 40.2. WDTIMER0_CONTROL Register Bit Descriptions

Bit	Name	Function
31:2	Reserved	Must write reset value.
1	DBGMD	**Watchdog Timer Debug Mode.** This bit determines the behavior of the watchdog timer when the device is halted during a debug operation. When cleared to 0, the watchdog timer continues to run during a debug halt (and will reset the device). When set to 1, the watchdog timer is halted during a debug halt operation.
0	EWIEN	**Early Warning Interrupt Enable.** 0: Disable the early warning interrupt (EWI). 1: Enable the early warning interrupt (EWI).

Notes:

1. Accessing any WDTIMER register takes several clock cycles of the associated low frequency oscillator. It will take at least 4 low frequency oscillator clock cycles to modify a bit and read back the modified value.
2. WDTIMER registers must be unlocked using the WDTKEY register in order to enable write access.

Register 40.2. WDTIMER0_STATUS: Module Status

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved											UPDSTS	RTHF	EWI	PRIVSTS	KEYSTS
Type	R											R	R	RW	R	R
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

WDTIMER0_STATUS = 0x4003_0010

This register also supports SET access at (ALL+0x4) and CLR access at (ALL+0x8)

Table 40.3. WDTIMER0_STATUS Register Bit Descriptions

Bit	Name	Function
31:5	Reserved	Must write reset value.
4	UPDSTS	**Watchdog Timer Threshold Update Status.** 0: An update completed or is not pending. The EWTH and RTH fields can be written. 1: An update of the threshold register is occurring. The EWTH and RTH fields should not be modified until hardware clears UPDSTS to 0.
3	RTHF	**Reset Threshold Flag.** 0: The counter is currently less than the reset threshold (RTH) value. 1: The counter is currently greater than or equal to the reset threshold (RTH) value.
2	EWI	**Early Warning Interrupt Flag.** This bit is set to 1 by hardware when an early warning match has occurred. Firmware may write the bit to 1 to manually trigger this interrupt. This bit must be cleared by firmware.
1	PRIVSTS	**Register Access Status.** 0: The watchdog timer registers are currently read-only. 1: A write transaction can be performed on the module registers.

Notes:

1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. Accessing any WDTIMER register takes several clock cycles of the associated low frequency oscillator. It will take at least 4 low frequency oscillator clock cycles to modify a bit and read back the modified value.
3. WDTIMER registers must be unlocked using the WDTKEY register in order to enable write access.

Table 40.3. WDTIMER0_STATUS Register Bit Descriptions

Bit	Name	Function
0	KEYSTS	**Key Status.** 0: No keys have been processed by the interface. 1: The attention key has been received and the module is awaiting a command.

Notes:
1. This register contains interrupt flags. Firmware should only use the SET and CLR addresses when modifying interrupt flags to avoid conflicts with hardware.
2. Accessing any WDTIMER register takes several clock cycles of the associated low frequency oscillator. It will take at least 4 low frequency oscillator clock cycles to modify a bit and read back the modified value.
3. WDTIMER registers must be unlocked using the WDTKEY register in order to enable write access.

Register 40.3. WDTIMER0_THRESHOLD: Threshold Values

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RTH															
Type	RW															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	EWTH															
Type	RW															
Reset	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Register ALL Access Address

WDTIMER0_THRESHOLD = 0x4003_0020

Table 40.4. WDTIMER0_THRESHOLD Register Bit Descriptions

Bit	Name	Function
31:16	RTH	**Reset Threshold.** When the counter value matches the value in RTH, the module will initiate a system reset if the watchdog timer is enabled as a reset source.
15:0	EWTH	**Early Warning Threshold.** When the counter value matches EWTH, the module will interrupt the core if watchdog timer early warning interrupt is enabled.

Notes:

1. Accessing any WDTIMER register takes several clock cycles of the associated low frequency oscillator. It will take at least 4 low frequency oscillator clock cycles to modify a bit and read back the modified value.
2. WDTIMER registers must be unlocked using the WDTKEY register in order to enable write access.

Register 40.4. WDTIMER0_WDTKEY: Module Key

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Reserved															
Type	R															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Name	Reserved								KEY							
Type	R								W							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register ALL Access Address

WDTIMER0_WDTKEY = 0x4003_0030

Table 40.5. WDTIMER0_WDTKEY Register Bit Descriptions

Bit	Name	Function
31:8	Reserved	Must write reset value.
7:0	KEY	**Watchdog Timer Key.** The watchdog timer key is used to perform certain functions on the watchdog timer, such as resetting the counter and locking the timer registers. The attention key (ATTN) must be written first, followed by a command key. The available commands keys are: WRITE, RESET, DISABLE, START, and LOCK. If the command key is not written within 32 APB clocks of writing the ATTN key, then the sequence must be started over with a new ATTN key write. 0xA5 : ATTN: Attention key to start the command sequence. 0xF1 : WRITE: Allow one write access to the WDT registers. 0xCC : RESET: Reset the WDT counter. 0xDD : DISABLE: Disable the WDT counter. 0xEE : START: Start the WDT counter. 0xFF : LOCK: Lock the WDT from any other writes until the next system reset. All other values are reserved.

Notes:

1. Accessing any WDTIMER register takes several clock cycles of the associated low frequency oscillator. It will take at least 4 low frequency oscillator clock cycles to modify a bit and read back the modified value.
2. WDTIMER registers must be unlocked using the WDTKEY register in order to enable write access.

40.8. WDTIMER0 Register Memory Map

Table 40.6. WDTIMER0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTIMER0_CONTROL	0x4003_0000	ALL | SET | CLR	Reserved																														DBGMD	EWIEN
WDTIMER0_STATUS	0x4003_0010	ALL | SET | CLR	Reserved																											UPDSTS	RTHF	EWI	PRIVSTS	KEYSTS

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

Table 40.6. WDTIMER0 Memory Map

Register Name	ALL Address	Access Methods	Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
WDTIMER0_THRESHOLD	0x4003_0020	ALL	RTH																EWTH															
WDTIMER0_WDTKEY	0x4003_0030	ALL	Reserved																								KEY							

Notes:

1. The "ALL Address" refers to the absolute address of the ALL access method for a register. A register may also support SET, CLR, and MSK access methods, as indicated by the "Access Methods" column. SET, CLR and MSK addresses are offset from the ALL address by 4, 8 and 12 bytes, respectively. For example, a register whose ALL address is located at 0x4001_00A0 in the address map may have a SET address at 0x4001_00A4, a CLR address at 0x4001_00A8, and a MSK address at 0x4001_00AC.

CONTACT INFORMATION

Silicon Laboratories Inc.

400 West Cesar Chavez
Austin, TX 78701

Please visit the Silicon Labs Technical Support web page:
https://www.silabs.com/support/pages/contacttechnicalsupport.aspx
and register to submit a technical support request.

Patent Notice

Silicon Labs invests in research and development to help our customers differentiate in the market with innovative low-power, small size, analog-intensive mixed-signal solutions. Silicon Labs' extensive patent portfolio is a testament to our unique approach and world-class engineering team.